

Reálné možnosti staganografických nástrojů

Real potential of steganography tools

Bc. Ludvík Vorbach

Diplomová práce
2011



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2010/2011

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Ludvík VORBACH**
Osobní číslo: **A09415**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**

Téma práce: **Reálné možnosti steganografických nástrojů**

Zásady pro vypracování:

1. Zpracujte rešerši týkající se problematiky steganografie v návaznosti na dostupné steganografické nástroje.
2. Charakterizujte základní nosná média běžně používaná pro přenos skrytých informací.
3. Proveďte rozbor steganografických algoritmů a otestování jejich účinnosti v digitální obrazové informaci.
4. Analyzujte zarušení jednotlivých steganogramů v závislosti na velikosti zprávy a použitém steganografickém nástroji.
5. Zhodnoťte zarušení digitálních obrazových dat pro jednotlivé steganografické nástroje.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. JOHNOSON, N.F. Information Hiding Steganography and Watermarking – Attacks and Countermeasures. Hardcover: Springer, 2000. ISBN 978-0-7923-7204-2.
2. KATZENBEISSER, S. Information hiding techniques for steganography and digital watermarking. Hardcover: Artech House Books, 2000. ISBN 1-58053-035-4.
3. COLE, E. Hiding in Plain Sight: Steganography and the Art of Covert Communication. Wiley, 2003. ISBN: 0471444499.
4. PRATT, W. Digital Image Processing, Wiley, 2007, ISBN 978-0-471-76777-0.
5. WAYNER, P. Disappearing Cryptography, Morgan Kaufmann, 2002, ISBN 1558607692.

Vedoucí diplomové práce:

Ing. Jiří Hološka

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

25. února 2011

Termín odevzdání diplomové práce:

27. května 2011

Ve Zlíně dne 25. února 2011

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

ABSTRAKT

Tato diplomová práce se zabývá problematikou steganografie a jejího reálného využití s běžně dostupnými steganografickými nástroji. Celá práce je rozdělena do dvou částí a to na část teoretickou a praktickou. Teoretická část diplomové práce popisuje historii a vývoj této vědní disciplíny. Následně je popsán steganografický proces a základní metody digitální steganografie. Na závěr popisují nejpoužívanější typy krycích medií a stegoanalýzu. Praktická část práce popisuje sedm steganografických systémů, jejich využití a testování na zkušební množině grafických souborů, navrnutí analyzačních postupů a aplikací. V závěru práce jsou zveřejněny výsledky v grafickém i písemném podání.

Klíčová slova: Kryptologie, steganografie, stegogram, stegoanalýza, krycí soubor

ABSTRACT

This thesis deals with steganography and its real use with commonly available steganography tools. It is divided into two parts, a theoretical and practical. The theoretical part of the theses describes the history and development of this discipline. Next description concerns steganography process and basic methods of digital steganography. Finally, I describe the most common covering media and steganalysis. The practical part describes the seven steganography systems, their use and testing on a test set of graphic file. I suggested by analyzing processes and applications. Results has been published in written and graphical form.

Keywords: Cryptology, steganography, stego-file, steganalysis, covering media

Touto cestou bych rád poděkoval vedoucímu své diplomové práce Ing. Jiřímu Hološkovi za odborné vedení, cenné připomínky a rady, které mi v průběhu realizace práce ochotně zprostředkoval. Především ale děkuji svým rodičům, za hmotnou a morální podporu po celé období studia.

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 POJEM STEGANOGRAFIE	11
1.1 KRYPTOLOGIE	11
1.1.1 Kryptografie	11
1.1.2 Kryptoanalýza	12
1.2 HISTORIE STEGANOGRAFIE.....	13
1.2.1 Cardanova mřížka	14
1.2.2 Agenturní systém	16
1.2.3 Neviditelný inkoust	16
1.3 MODERNÍ STEGANOGRAFIE	17
1.3.1 Steganografický proces	18
1.3.2 Steganografie s využitím kryptografie	19
1.3.3 Základní aplikace steganografických metod	20
1.3.4 Steganografické metody.....	21
1.3.4.1 LSB (least significant bit) na nekomprimovaném obrazovém coveru	21
1.3.4.2 LSB na JPEG	24
1.3.4.3 Metoda rezervních dat - TCP	26
1.3.4.4 Metoda rezervních dat – MP3.....	27
1.3.5 Nejpoužívanější steganografická nosná media současnosti.....	30
1.4 STEGOANALÝZA	36
II PRAKTICKÁ ČÁST	40
2 STEGANOGRAFICKÉ NÁSTROJE	41
2.1 JP HIDE AND SEEK	42
2.2 STEGHIDE	43
2.2.1 Steghide v příkazovém řádku	44
2.3 OPENPUFF V3.10 STEGANOGRAPHY & WATERMARKING	45
2.4 INVISIBLE SECRETS 4	46
2.5 STEGANOS	48
2.6 HIDE IN PICTURE	49
2.7 S – TOOLS	50
3 TESTOVÁNÍ	52
3.1 METODY ANALYZAČNÍCH POSTUPŮ	52
3.2 ANALYZAČNÍ SOFTWARE.....	55
3.2.1 Bits Comparator	56
3.2.1.1 Ukázka zdrojového kódu	58
3.3 JPEG.....	61
3.3.1 Zhodnocení JPEG.....	63
3.4 BMP	67
3.4.1 Zhodnocení BMP	69
3.5 VYHODNOCENÍ PROGRAMŮ	73
ZÁVĚR	74

ZÁVĚR V ANGLIČTINĚ.....	75
SEZNAM POUŽITÉ LITERATURY.....	76
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	79
SEZNAM OBRÁZKŮ	80
SEZNAM TABULEK.....	82
SEZNAM PŘÍLOH.....	83

ÚVOD

Myšlenka a snaha utajit nebo skrýt komunikaci před zraky možných nepřátel či jen zvědavců je zde od nepaměti. Už v době paleolitu, kdy naši předkové obývali planetu Zemi, byla potřeba utajit určité informace před zneprátenou tlupou, nebo přijít na způsob dorozumívání při lovu tak, aby kořist lovců neslyšela. Stejně tak i dnes, v době moderní techniky s velkým množstvím komunikačních kanálů je kladen velký důraz na ochranu citlivých dat, soukromí a informací týkajících se jedinců nebo celé skupiny lidí. Velkého rozmachu se v této oblasti dosáhlo ve 20. století s rozvojem elektroniky a elektrotechniky. Taková vědní disciplína, která přetváří data do nesrozumitelného kódu a k jehož zpětné transformaci je zapotřebí znát tajný klíč, se nazývá kryptologie. Ta v sobě skrývá dva ucelené obory. Kryptografii a méně známou steganografii.

Steganografie se zabývá ukrytím komunikace. Ta je v dnešní době na okraji pozornosti odborné společnosti, což je velká škoda. Většina populace, jež chrání svá data, využívají služby kryptografie. Tedy svá data převede pomocí různých metod na shluk symbolů, které jsou nečitelné. Ale každý kdo se k takovým to datům dostane, hned pozná, že je zde ukryto něco cenného, něco co stojí majiteli za ochranu. Naproti tomu pokud jsou data schována pomocí nějaké ze steganografických metod, málokoho napadne, že za zdánlivě fádni obálkou se skrývají tajemství.

Internet ve spojení s výpočetní technikou má v současné době své nezastupitelné místo. Pro desítky miliónů lidí je toto spojení hlavním komunikačním kanálem a jeho využití je na každodenním pořádku. S tím jak se množství vyměněných informací a dat na internetu zvětšuje, roste i riziko ztráty dat, odcizení či jeho úmyslné zneužití. Steganografie tedy tvoří zajímavý prostředek jak svá data při komunikaci chránit. Navíc drtivá většina steganografických systémů podporuje i některou z kryptografických šifer. Touto kombinací se Vaše data stanou takřka nedosažitelná. Proto jsem zvolil toto téma diplomové práce, která nás přehledně provede historií steganografie, i jejím moderním pojetím. Nahlédneme na jednotlivé steganografické metody a podíváme se na současné druhy krycích medií. Nadále se seznámíme s některými vybranými aplikacemi, které využívají pro tajnou komunikaci digitální obrazové formáty JPEG a BMP.

I. TEORETICKÁ ČÁST

1 POJEM STEGANOGRRAFIE

Steganografie patří do vědního oboru kryptologie. Název steganografie je řeckého původu a vychází ze slov *steganos* (schovaný) a *graphein* (psát). Jak je již očividné z překladu, jedná se o disciplínu zabývající se utajenou komunikací pomocí ukrytí zprávy. Úkolem je tedy předávanou zprávu, která je zpravidla psána ve srozumitelné podobě, ukryt takovým způsobem, aby ji případný útočník nerozeznal.

1.1 Kryptologie

Slovem kryptologie označujeme skupinou vědních oborů zabývající se problematikou utajení obsahu zprávy. Kryptologie se skládá ze tří vědních disciplín a to kryptografií, kryptoanalýzou a již zmiňovanou steganografií.

1.1.1 Kryptografie

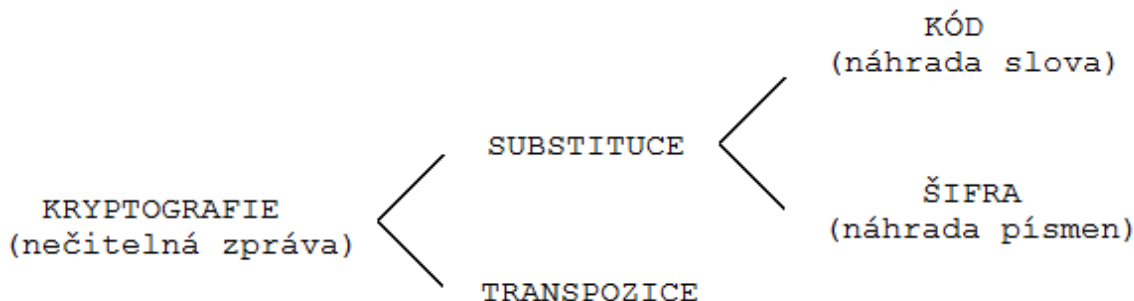
Kryptografie se zabývá matematickými metodami se vztahem k takovým prvkům informační bezpečnosti, jako je zajištění důvěrnosti zprávy, integrity dat (neporušenosti), autentizace entit (ověřování subjektu) a původu dat (vlastnictvím) – včetně zkoumání jejich silných stránek a slabín i odolnosti vůči různým metodám útoků. Ve starším chápání to byla věda o tom, jak navrhovat a používat šifrovací systémy, a tedy disciplína, která se zabývá převedením informace do podoby, v níž je tato informace skryta. Jejím úkolem bylo tedy učinit výslednou zprávu neurčitou i v situacích, kdy je plně prozrazený, zachycená třetí – nepovolanou stranou. Tím se liší od steganografie, jejímž úkolem je skrýt samotnou existenci zprávy, ale zpráva samotná může být napsána nebo předána ve srozumitelné podobě.¹

Metody používané v kryptografii můžeme rozdělit do dvou větví – transpozici a substituci. Při transpozici se písmena zprávy uspořádají jiným způsobem než původním, jde tedy vlastně o přesmyčku. Takový postup není příliš bezpečný u velmi krátkých zpráv, je tedy vhodné volit vhodnou délku zprávy. Alternativou k transpozici je substituce. Zde se jedná o techniku spočívající v náhodném spárování písmen abecedy a poté nahradit každé písmeno původní zprávy jeho partnerem. Jednoduše lze popsat rozdíl mezi transpozicí a

¹ VONDRUŠKA, P. *Kryptologie, šifrování a tajná písma*. Praha: Albatros, 2006. 8 s.

substitucí takto: „U transpozice si písmena zachovávají svoji identitu, ale změni pozici, u substituce je tomu přesně naopak“.²

Obrázek číslo jedna ukazuje dělení kryptologie.



Obr. 1 – Rozdělení kryptografie

Kryptografové jsou ti, kteří se zabývají návrhem, používáním a zkoumáním šifrovacích systémů a dalších aspektů informační bezpečnosti. V minulosti byl hlavním cílem kryptografie rozvoj algoritmů (postupů), které lze použít ke skrytí obsahu zprávy před všemi s výjimkou vysílací a přijímací strany (utajení); později přibyl rozvoj algoritmů sloužících k jednoznačnému určení osoby odesílatele (identifikaci) a k ověření správnosti zprávy přijímací stranou (autentizaci) a další související algoritmy.¹

1.1.2 Kryptoanalýza

Kryptoanalýza je jakýsi „opak“ kryptografie. Z toho plyne, že jedním z hlavních cílů je studium metod luštění šifrovacích systémů. Obecněji se kryptoanalýza zabývá analýzou odolnosti (síly) kryptografických systémů; a hledá metody vedoucí k proniknutí do těchto systémů.¹

¹ VONDRUŠKA, P. *Kryptologie, šifrování a tajná písma*. Praha: Albatros, 2006. 9 s.

² SINGH, S. *Kniha kódů a šifer „Utajování od starého Egypta po kvantovou kryptografii“*. Praha: Dokořán a Argo, 2009. 21,23,24 s.

Ten, kdo se zabývá kryptoanalýzou, je kryptoanalytik. Kryptoanalytici se snaží získat ze zašifrovaných zpráv její původní podobu – otevřený text. Cílem však může být i získání alespoň části skrytých informací.¹

1.2 Historie steganografie

První zmínky o steganografii jsou přibližně 2500 let staré. Jednalo se o Řeka jménem Demaratus, který poslal varování do Sparty před nebezpečím útoku Xerxovou armádou. V oněch dobách nebyl znám papír, a proto byl text zaznamenáván na dřevěné destičky s vrstvou vosku. Demaratus tento vosk odstranil a varování vyryl přímo do dřeva. Poté na destičku nanesl vosk zpět. Tabulka byla k nerozeznání od normální nepopsané tabulky, proto nevzbudila u strážce žádné podezření.

Na další formou ukrytí přišel Histiaios. Ten k odeslání zprávy využil hlavu svého sluhu. Oholil ji a na kůži vytetoval zprávu určenou Aristagoru Milétskému. Počkal, až vlasy dorostou a pak sluhu vyslal na cestu. Zpráva obsahovala popud proti perskému králi.

Postupem času se objevovaly stále rafinovanější formy ukrytí zprávy. Číňané psali zprávy na tenké hedvábí, které zmačkali do malé kuličky a zalili voskem. K přenosu zprávy docházelo prostřednictvím posla, který tuto kuličku požířel. Jiný způsob praktikoval Ital Giovanni Porta v 16. století. Jeho zpráva byla ukryta ve vejci vařeném natvrdo, na které psal speciálním inkoustem zhotovený z jedné unce kamence a pinty octa. Póry na povrchu skořápky umožní inkoustu proniknout do bílku, kde zanechá nesmazatelnou stopu. Zprávu lze tedy přečíst po oloupání vejce.

Další forma steganografie, která stojí za zmínění, byla používána v letech 1941. Jednalo se o mikrotečky, tedy fotograficky zmenšená stránka textu do tečky o průměru menším než milimetr. Tato „nevinná“ tečka byla použita v dopisu jako běžná tečka za větou. Zprávu lze přečíst pomocí mikroskopu. Na tuto formu ukrytí zprávy přišli Němečtí agenti působící v Latinské Americe.

Během staletí byly použity stovky méně či více zdařilých způsobů utajení zpráv.

¹ VONDRUŠKA, P. *Kryptologie, šifrování a tajná písma*. Praha: Albatros, 2006. 9 s.

Vyjmenujme zde jen nepoužívanější:

- použití neviditelného inkoustu;
- vyznačení písmen v jinak nezávadném textu;
- některá písmena v nezávadném textu byla propíchnána špendlíkem, tato písmena tvořila předávaný utajený text;
- podobně jsou některá písmena v jinak nezávadném textu psána např. tučněji, jiným sklonem, jinou velikostí;
- ukrytí znaků otevřeného textu na domluvených místech;
- prvá (druhá, poslední) písmena některých domluvených slov v dopise tvoří krátký utajený text;
- text je napsán na předem domluvená místa a je obklopen nezávadným textem, příjemce přiloží dešifrovací tabulku a přečte si předávaný text;
- zápis šachové partie;
- notový zápis;
- návod na vaření, háčkování;
- mikrotečky;
- digitální steganografie.¹

O účinnosti steganografie se můžete sami přesvědčit a to v prvním odstavci této kapitoly (Historie steganografie). Budete-li číst pozorněji, můžete se všimnout, že některá písmena jsou menší. Utajený text zní „jsem prozrazen, končím tu“.

Podrobněji se podíváme jen na tři nejpoužívanější systémy důmyslného ukrytí zpráv. Cardanovu mřížku, agenturní systém a neviditelný inkoust.

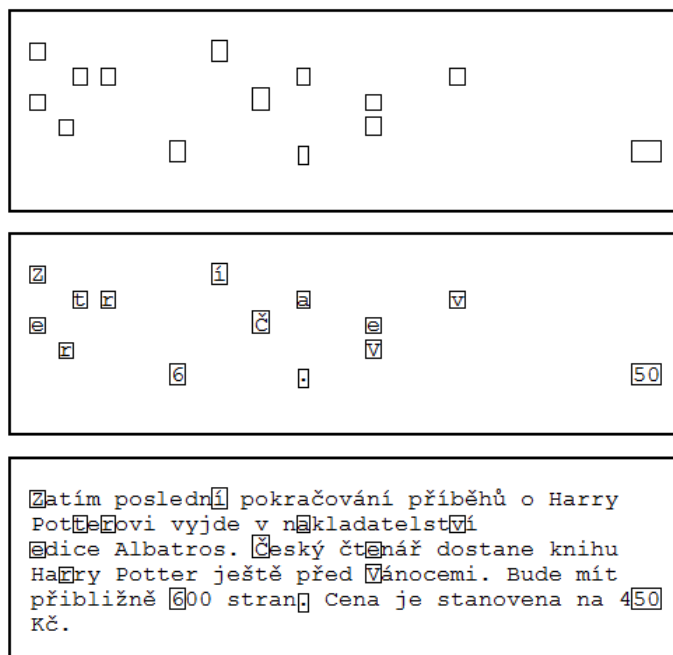
1.2.1 Cardanova mřížka

Jedná se o jednu ze starších steganografických metod, která byla hojně využívána kvůli své jednoduchosti a účinnosti. Za návrhem stojí italský fyzik, matematik, lékař, astronom a

¹ VONDRUŠKA, P. *Kryptologie, šifrování a tajná písma*. Praha: Albatros, 2006. 127 s.

filozof Girolamo Cardano narozen 24. září 1501. Princip spočívá v obdélníkové mřížce, ve které jsou na určitých pozicích zhotovené otvory. Tuto mřížku přiložíme na list čistého papíru a do otvorů vepisujeme jednotlivá písmena skrývaného textu. Po vyplnění mřížku sejme a písmena vhodně doplníme textem, tak aby výsledný text nevzbuzoval podezření. Pro dešifrování příjemce pouze přiloží svou tabulku se stejným rozmístěním otvorů a ukrytou zprávu jednoduše přečte.

Tento způsob je velice efektivní za předpokladu, že mřížka neobsahuje velké množství otvorů. V opačném případě by bylo velice obtížné doplnění písmen tak, aby výsledný text byl smysluplný. Stejně tak je nevhodné doplnění zcela náhodnými písmeny popř. symboly. Došlo by k vytvoření souboru, který je reprezentován stochastickým shlukem symbolů a je tedy na první pohled podezřelý z možností obsahu skryté zprávy.



Zatím poslední pokračování příběhů o Harry Potterovi vyjde v nakladatelství edice Albatros. Český čtenář dostane knihu Harry Potter ještě před Vánoce. Bude mít přibližně 600 stran. Cena je stanovena na 450 Kč.

Obr. 2 – Cardanova mřížka

Na obrázku dvě je praktická ukázka Cardanovy mřížky. Otevřený text zní „zítra večer v 6.50“.

1.2.2 Agenturní systém

Agenturní systém byl velice oblíben v době studené války a to na obou stranách železné opony. Opět se jedná o typického zástupce steganografických textových metod. Prokazatelně byl využíván ještě v dobách totality při komunikaci mezi agenturní sítí v České republice a jejími řídicími centry v cizině. Tehdy měl podobu nenápadných dopisů odesílaných z ČR na smluvené adresy v cizině (zejména ve Spolkové republice Německo). V textu dopisu bylo na domluveném místě (např. konec druhého slova v každé větě, poslední slovo věty apod.) písmeno otevřeného textu (ve složitějších variantách zde byly místo písmene otevřeného textu souřadnice kódové tabulky, kterou obě strany používaly). Charakteristická pro takto vytvořené texty byla jejich „kostrbatost“ a nepřírozená slovní zásoba. Hlavní výhodou bylo to, že se při běžném zhlédnutí textu utajilo odeslání senzitivní informace. Při zaslání šifrovaného textu by byl v té době odesílatel automaticky podezřelý z podvratné činnosti. Při přesném dodržení různých provozních pravidel (krátký otevřený text; použití kódové tabulky, která se pravidelně obměňuje; nepřilíš častý provoz; „chytrá“ slohová a obsahová stavba odeslaného dopisu) může být systém relativně bezpečný.¹

Příklad:

Otevřený text: Kdo jinému jámu kopá, sám do ní padá.

Pravidlo pro utajení textu: Otevřený text tvoří prvá písmena slov odeslaného textu.

Výsledný text: Karel dnes odjížděl. Jindřich i Nataša Esterová mu uvařili jídlo. A moc už k odjezdu pospíchal a spěchal. A musel docela okamžitě, nikoliv ideálně, pobalit a dát ádíjé.¹

1.2.3 Neviditelný inkoust

Neviditelný, též sympatetický inkoust je velmi používanou metodou steganografie. První zmínky na výrobu takového inkoustu jsou z prvního letopočtu našeho století. Plinius Starší zjistil jak vhodně využít pryšce, slovensky též mliečnika (Tithymalus sp. z čeledi

¹ VONDRUŠKA, P. *Kryptologie, šifrování a tajná písma*. Praha: Albatros, 2006. 131 s.

Euphorbiaceae). Text, který měl být určen pouze příjemci, byl napsán mlékem, které vytéká ze stonku. Po úplném zaschnutí je text průhledný. Prosté zahřátí způsobí rozklad organických látek (cukrů, popř. bílkovin), což zapříčiní zhnědnutí mléka a text se tak stane čitelný.

Nezpochybnitelnou výhodou neviditelného inkoustu je absence složitého školení při jeho používání, proto je používán i dnes. Jsou zaznamenány případy komunikaci vězňů s okolním světem prostřednictvím neviditelného inkoustu zhotoveného z vlastní moči. Moč při zahřátí též zhnědne.

Moderní steganografie zná hned několik principů, na kterých fungují sympatetické inkousty. Jde o již dvakrát zmiňovaný způsob zahřátí organických látek a jeho následného rozkladu. Zde jsou zástupci moč, mléko, citronové šťávy, ovocné a octové šťávy. Ve většině případů je výsledný text hnědý, můžeme však dosáhnout i žlutých, zelených černých a modrých odstínů. Druhou skupinou jsou inkousty založené na chemické reakci dvou látek. Zde je nepřehledné množství vzájemně reagujících látek, které dokážou vytvořit černé, hnědé, modré a červené odstíny inkoustu. Poslední běžnou metodou je využití fluorescenčního jevu. Tedy jevu, kdy látka pohltí záření o vyšší energii a vyzáří světlo nižší energie. Jako činitel pro zviditelnění zprávy je použito ultrafialové popřípadě infračervené záření. Typický zástupce takovéto fluorescenční látky je kyselina salicylová, která se stane viditelná pouze pod UV zářením.

1.3 Moderní steganografie

Dvacáté století lze charakterizovat jako století páry, velkého množství vojenských konfliktů, elektřiny a nepředstavitelného rozmachu v oblasti elektroniky. Za tento úžasný boom vděčíme zčásti právě vojenským konfliktům respektive vojenskému výzkumu, který měl za úkol vývoj mnoha strategických technologií. Zejména se jednalo o komunikační systémy. Paralelně s vývojem těchto systémů rostla potřeba zpracování stále většího objemu dat, což obstarávala výpočetní technika reprezentovaná počítači. Jelikož se nacházíme ve vojenské sféře, je zde značná potřeba veškerou komunikaci zabezpečit proti neoprávněnému zneužití. Toto nebezpečí má za úkol eliminovat kryptografie a steganografie, které dostaly s využitím IT technologie zcela nový rozměr. V tomto období dochází v celém vědním oboru kryptologie k obrovskému rozvoji.

Pojem digitální steganografie, který se objevuje v mnoha publikacích a člancích, je tedy logicky spjat s digitálně zpracovanými daty. Nejčastěji se jedná o obrázek, audio či video záznam, text nebo metadata různých komunikačních protokolů (např. TCP/IP). Toto jsou v dnešní době nejčastěji využívaná media pro aplikaci steganografie.

1.3.1 Steganografický proces

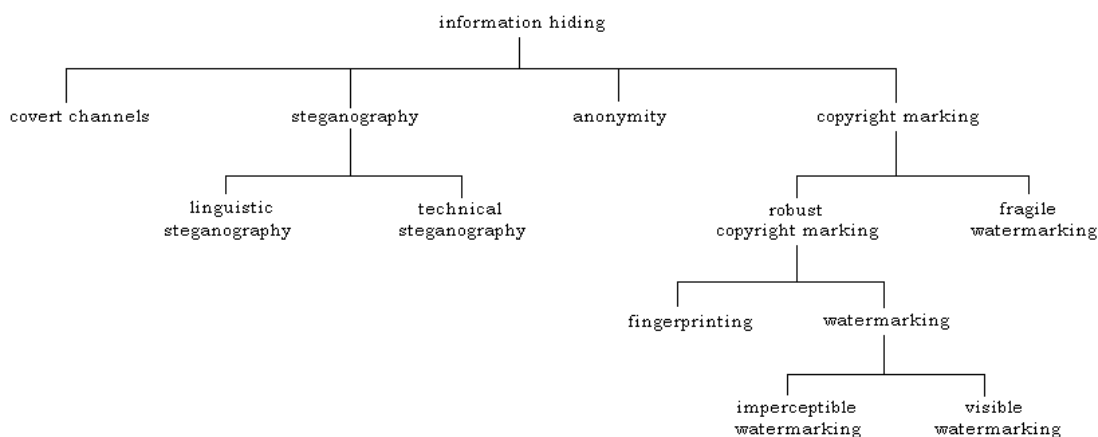
Nosná media, do nichž vkládáme tajnou zprávu, jsou hromadně označovány termínem *cover*, někdy též označované jako *covering medium*, popřípadě *carrier medium*. Pod termínem *steganogram* (zkr. stegogram) se dále, jak intuice napovídá, chápe instance coveru, v níž je ukryta tajná zpráva (*embedded*, *embedded message*, *hidden message*, *embedded data*).

Cover + embedded (+stego-key) = steganogram

U některých steganografických softwarů se můžeme setkat s použitím hesla (stego-key), což má za účel zvýšit bezpečnost komunikace. Bez znalosti tohoto klíče není druhá strana schopná extrahovat utajenou zprávu z coveru.

Maximální velikost tajné zprávy, kterou lze daným stegosystémem do konkrétního coveru vetknout, se nazývá *steganografická kapacita* tohoto coveru. Steganografie se někdy považuje pouze za jednu z několika disciplín spadající pod tzv. *information hiding*.

Zevrubnou oborovou strukturu znázorňuje následující strom v obrázku číslo 3.⁴



Obr. 3 – Struktura „information hiding“

⁴ ŽILKA, R. *Steganografie a stegoanalýza* [online]. 2008. Diplomová práce. 10 s.

Zatímco steganografii jako takové se dostává zájmu z okruhů akademických, vojenských, vládních a policejních, stěžejní problematikou v komerční sféře je vodotisk (*watermarking*). Je to zřejmé již z nákresu, kde se příslušný uzel nachází v kategorii *copyright marking*. Rozdíl mezi steganografií a vodotiskem spočívá v kladení důrazu na různé kvalitativní aspekty ukrytí dodatečné informace do coveru: pro steganografii je podstatné, aby přítomnost této informace nebyla nápadná; pro vodotisk, aby dodatečná informace nebyla z coveru odstranitelná (přesněji řečeno, aby její odstranění nutně vedlo v jistém smyslu k destrukci či naprostému znehodnocení coveru) a na její detekovatelnosti buď nezáleží, nebo je přímo nezbytná.⁴

Vodotisk tedy klade důraz na robustnost zatímco steganografie na objem (utajených) dat v coveru. I pro samotnou steganografii je jistě vhodné, aby do coveru dané velikosti bylo možné vložit co nejvíce dat tak, že ani poškození stegogramu tajná data nezničí. Existuje však jistá (nejasná) míra bitového objemu, kterou nesmí vkládaný objekt překročit, aby výsledný stegogram stále vypadal dostatečně nenápadně. Tento bitový objem zahrnuje jak samotnou utajovanou zprávu, tak režijní data stegosystému a redundanci, která zajistí robustnost. Z uvedeného textu vyplývá stěžejní invariant platný obecně pro všechny stegosystémy:

utajovaná zpráva [b] + redundantní informace vynaložená na robustnost [b] = \underline{k}

Kde \underline{k} je konstanta daná coverem a očekávaným stegoanalytickým algoritmem. Tento vztah v sobě nese onu relativitu představy „stegogram vypadající nenápadně,“ neboť je úměrný právě očekávanému stegoanalytickému útoku. Proto nelze \underline{k} nazývat steganografickou kapacitou, která je jasně deterministicky vyčíslitelná z coveru a stegosystémem (nehledě na případnou metodu stegoanalýzy).⁴

1.3.2 Steganografie s využitím kryptografie

Z výše uvedených řádků mnohým vyvstane otázka, zda steganografie využívá služby kryptografie. O odpověď zní, zcela jednoznačně. V moderním světě, kde internet tvoří pro mnohé jedince a společnosti stěžejní komunikační prostředek je využití šifrování při steganografii zcela běžné. Celosvětový rozvoj IT technologie a rozšíření sítě internet do

⁴ ŽILKA, R. *Steganografie a stegoanalýza* [online]. 2008. Diplomová práce. 10-11 s.

všech koutů světa přináší i mnohá nebezpečí spojená s velkým počtem osob, které cíleně získávají citlivá data prostřednictvím těch nejsofistikovanějších postupů. Proto je potřeba svá cenná data ochránit všemi dostupnými způsoby. Navíc kombinace těchto kryptologických metod, činí jejich prolomení a rozluštění takřka za nemožné. Nevýhodou může být fakt, že šifrovaná data mají tendenci tvořit binární řetězce, které pachatelé mohou poukázat na skrytá data. To lze však vyřešit použitím stegosystému, které vhodně rozmístí utajovaná data tak, že se pozorovateli jeví jako šum. V neposlední řadě lze využít obou variant kryptografie. Myslím tím symetrické respektive asymetrické šifrování. Použitím těchto dvou metod je vázáno na problematiku výměny a správy klíčů, což není oblast, kterou se diplomová práce bude zabývat.

1.3.3 Základní aplikace steganografických metod

Injection steganography method – tato metoda vkládá utajená data do nosného media takovým způsobem, že výsledný stegogram se ve své prezentaci neliší od originálního coveru. Tedy skrytá data jsou uložena tak, aby žádným způsobem neovlivnila cover. Nevýhodou je změna velikosti originálního coveru oproti stegogramu, rozdíl velikostí tvoří velikost utajované zprávy a režijní data. Tím je dán stěžejní požadavek, tedy že nosné medium nesmí mít pevnou délku (velikost). Nejvhodnějšími zástupci jsou některé formáty obrázků, audio či video, HTML kód a datagramy.

Substitution steganography method – substitution neboli nahrazení. Již toto slovo samotné napovídá, že při této metodě budeme některá data nahrazovat daty, která chceme ukrýt. Otázkou je, která? Jak se dočteme později nerozšířenější metodou steganografie vůbec, je právě substituční metoda známá pod zkratkou LSB (least significant bit) neboli metoda nejméně významných bitů. Tím však dochází k degradaci coveru. Tato diference ale nemusí být detekována, obzvláště v případech, kdy se do hry zapojí naše ne zcela dokonalé smysly. Protože ukrytá data zaměňujeme, nikoliv přidáváme, celkový datový objem coveru se neliší od výsledného stegogramu. Jednotná délka nám však limituje délku (objem) ukrývané zprávy. Pokud by velikost nebyla volena smysluplně, nastane nepřijatelné zkreslení coveru a skrytý přenos může být prozrazen. Nejpoužívanějšími nosnými medii je v tomto případě opět obrazový formát, datagramy síťových protokolů, audio, video a souborové systémy.

Propagation steganography method – obě výše zmiňované metody potřebují pro svoji úspěšnou aplikaci dva druhy souborů. Tajnou zprávu a nosné médium. Tato metoda je

výjimečná tím, že k utajení zprávy nosné médium nepotřebuje a to z toho důvodu, že si sama cover vytvoří. Celá filozofie tohoto systému je postavena na myšlence, že každá utajovaná zpráva může mít nosné medium ušito přímo na míru své potřebě. Tedy tak, aby se výsledný stegogram tvářil co nejpřirozeněji a byl co nejmenší. Výstupem tedy může být například fraktální obrazec s barvou, křivkami i velikostí dle potřeby utajované zprávy.

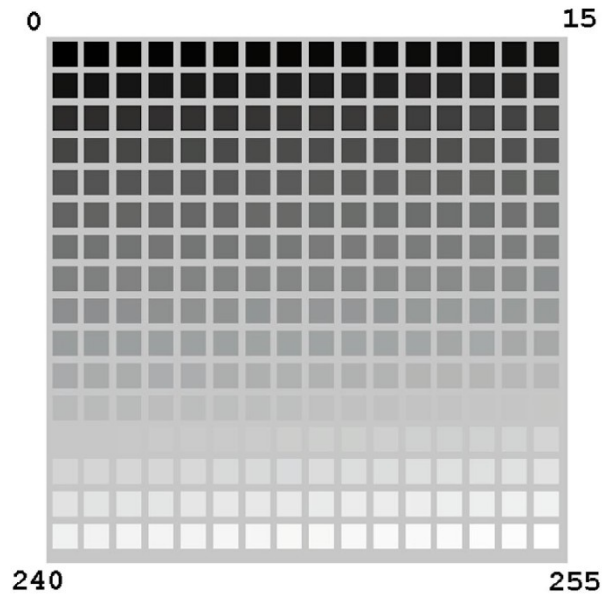
1.3.4 Steganografické metody

V této kapitole uvedu pouze dvě nejpoužívanější steganografické metody a to metodu využití LSB a metodu využití rezervních dat. Některé literatury uvádí jako další metodu, využití rozprostřeného spektra a metodu způsobu zápisu. Tyto dvě metody jsou však na okraji pozornosti, proto se jim více věnovat v této práci nebudu.

1.3.4.1 *LSB (least significant bit) na nekomprimovaném obrazovém coveru*

Jak je již z názvu této steganografické metody zřejmé, ukryvaná data jsou zapisována do pozice nejméně významného bitu. To nám zaručí minimální změnu coveru oproti originálu. Tuto metodu lze aplikovat s menší či větší úspěšností na veškeré binárně vyjádřené data, u kterých není překážkou přepisování jednotlivých bitů.

Pro názornou ukázkou jsem zvolil neztrátový obrazový formát BMP. Mějme obrázek o velikosti 1024x768 pixelů. Každý bod je zde reprezentován 24 bity, což představuje standardní barevnou hloubku. Těchto 24 bitů reprezentuje tři základní barevné složky, tedy RGB (red, green, blue). Tzn., že každá jednotlivá barva je vyjádřena 8 bity = 1 byte (bajt). Řečí čísel mluvíme o $2^8 = 256$ (0 - 255) barevných odstínů jedné barevné složky. Tato paletu zobrazuje obrázek číslo čtyři.



Obr. 4 – Standardní paleta 256 odstínů šedé

Binární vyjádření	Hexadecimální vyjádření	Výsledná barva
00000000 00000000 11111111	00 00 FF	modrá
00000000 11111111 00000000	00 FF 00	zelená
11111111 00000000 00000000	FF 00 00	červená
00000000 00000000 00000000	00 00 00	černá
11111111 11111111 11111111	FF FF FF	bílá

Tab. 1 – Barvy a jejich číselné vyjádření

$$1024 \times 768 = 786\,432 \text{ celkový počet pixelů}$$

$$786\,432 \times 24 = 18\,874\,368 \text{ celkový počet bitů}$$

$$18\,874\,368 \div 8 = 2\,359\,296 \text{ všechny nejméně významné bity}$$

$$2\,359\,296 \div 8 = 294\,912 \text{ všechny nejméně významné bajty (byte)}$$

$$294\,912 \div 1024 = 288 \text{ kB}$$

Tedy to znamená, že do takového obrázku, jehož velikost je cca 2,25MB lze vložit až 288 kB tajných dat. Další možností jak zvětšit, už tak podle mě dostatečné místo na ukrytí zprávy je přepisovat nikoliv poslední jeden bit, ale třeba poslední dva či tři. Tento krok však může mít za následek zkreslení obrázku až do takové míry, že odhalení tajné komunikace je více než jisté. Vhodnější metodou jak umožnit bezpečné doručení tajné zprávy bude bezpochyby zvolení jiného steganografického media.



Obr. 5 – Ovlivnění kvality obrázku ukládáním bitů na vyšší pozici v bajtu – dostupný z <http://adela.utko.feec.vutbr.cz/mbis/prednaska/MBIS%2013.pdf>.

Zde je ukázka vložení písmene A, které je ASCII kódem v binární podobě vyjádřeno jako 01000001 do matice 3x3 bajty (pixely). Bity na pozici LSB jsou podtržené.

1100101 <u>1</u>	0010100 <u>1</u>	0110110 <u>0</u>
0010111 <u>1</u>	1000101 <u>1</u>	0000011 <u>0</u>
0111100 <u>1</u>	1011110 <u>1</u>	1000000 <u>0</u>

Nyní do této matice zakomponujeme námi ukrývaný znak A.

1100101 <u>0</u>	0010100 <u>1</u>	0110110 <u>0</u>
0010111 <u>0</u>	1000101 <u>0</u>	0000011 <u>0</u>
0111100 <u>0</u>	1011110 <u>1</u>	1000000 <u>0</u>

Červeně jsou vyznačeny ty bity, jejichž hodnota se musela změnit (metoda negace respektive inkrementace o 1), aby byla výsledná binární sekvence obsahovala utajený symbol. Modře jsou vyznačeny bity, jejichž hodnota zůstala shodná s originálem.

Slabou stránkou této metody aplikovanou na bezztrátové obrazové formáty je nemožnost sebemenší úpravy obrázku, který již obsahuje utajenou zprávu. Například použití jakéhokoliv filtru, změna velikosti, komprimace či změna rozlišení znamená, že velké množství bitů na nejméně významné pozici se změní, čímž je ukrytá zpráva degradována.

Řešením tohoto problému, je využít celý prostor nosného media pro embedding. Jako nejjednodušší metodu lze zvolit opakování. Utajená zpráva se do coveru vloží několikrát, čímž se zvýší pravděpodobnost, že po průchodu filtrovacím algoritmem zůstane zpráva neporušená. Abychom neukládali opakované zprávy hned po sobě, využijeme generátor pseudonáhodných čísel, kterým docílíme vhodnějšího rozložení utajené zprávy do celého coveru.

Pokročilejší algoritmy využívající metodu LSB nevybírají bajty, do kterých se bude utajená informace ukládat nahodile, ale cíleně. Celý obraz je nejdříve analyzován a jsou porovnávány jednotlivé pixely a jejich okolí. V případě, že okolní body se výrazným způsobem neliší, nejsou vhodným kandidátem na přepsání (např. jasné modré nebe). Naopak algoritmus ukládá utajenou informaci do míst, kde se okolní body značně liší a naše oko v těchto místech nerozezná určitou diferenci barev (hrany a linie objektů, výrazné barevné přechody).

Jiným způsobem může být rozptřetí informace do statistiky jasu pixelů - systém může náhodně vybrat několik dvojic pixelů, u nichž zvýší nebo sníží kontrast jasu - průměrný jas obrázku se tím nezmění, ale zvýší se kontrast v rámci množiny vybraných pixelů. Tato metoda údajně při vhodné volbě parametrů vydrží kompresi používanou formátem JPEG.⁷

1.3.4.2 LSB na JPEG

V případě, že jako krycí medium nepoužijeme nekomprimovaný obrazový formát, který kóduje bit po bitu, jako např. BMP, GIF, TIFF ale ztrátový obrazový formát JPEG (dnes

⁷ RYŠÁKOVÁ, A. *Steganografie 1* [online]. 2003.

nejrozšířenější grafický formát na světě), bude problematika ukrytí tajné informace o trochu složitější.

Komprimované kódování u JPEG formátu je realizováno následnými kroky. Nejdříve se celý obrázek převede z barevného prostoru RGB do YCbCr (Y – jasová složka, Cb a Cr – modrá a červená chrominanční složka). Důvod této úpravy tkví v odstranění některých redundantních dat (snížení korelace mezi jednotlivými složkami) a odlišení jasové složky, která nese pro oko důležitější informace. Nyní se celá bitmapa rozdělí na čtverce o rozměrech 8x8 pixelů, které jsou následně podrobeny diskrétní kosínové transformací (DCT - discrete cosine transform). Výsledkem této operace je převedení původních 64 hodnot, jež reprezentovala jasovou a barevnou složku na 64 nových DCT koeficientů reprezentující daný čtverec.

$$Y_{u,v} = \alpha(u) \times \alpha(v) \times \sum_{m=0}^7 \sum_{n=0}^7 x_{m,n} \times \cos \left[\frac{\pi}{8} \left(m + \frac{1}{2} \right) u \right] \times \cos \left[\frac{\pi}{8} \left(n + \frac{1}{2} \right) v \right]$$

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{8}} & \text{pro } k = 0 \\ \sqrt{\frac{2}{8}} & \text{pro } k > 0 \end{cases}$$

Ve třetím kroku se provádí kvantizace těchto koeficientů a to následujícím způsobem. Původní hodnota je vydělena určitou konstantou. Tato konstanta závisí na poloze koeficientu v bloku a pro nižší koeficienty je nižší (v obvyklém případě). Dále se liší v závislosti na barevné složce obrázku (menší ztráta pro jasovou složku než pro barevné složky). Hodnoty těchto konstant jsou uloženy v kvantizační tabulce v ukládaném souboru a mohou se tedy pro každý obrázek lišit.⁹

Právě po tomto třetím kroku, se aplikuje metoda LSB, která nám do JPEGu ukryje námi požadovaná data. Změna jednoho bitu, tedy ovlivní všechny bity v daném bloku. Z logiky věci by snaha o vložení ukryvaných dat, kdykoliv před kvantováním mohla skončit degradací popřípadě úplnou ztrátou skrytě přenášené zprávy.

⁹ ONDRUŠ, J. *Beztrátová komprese JPEG grafiky* [online]. 2009. 7 s.

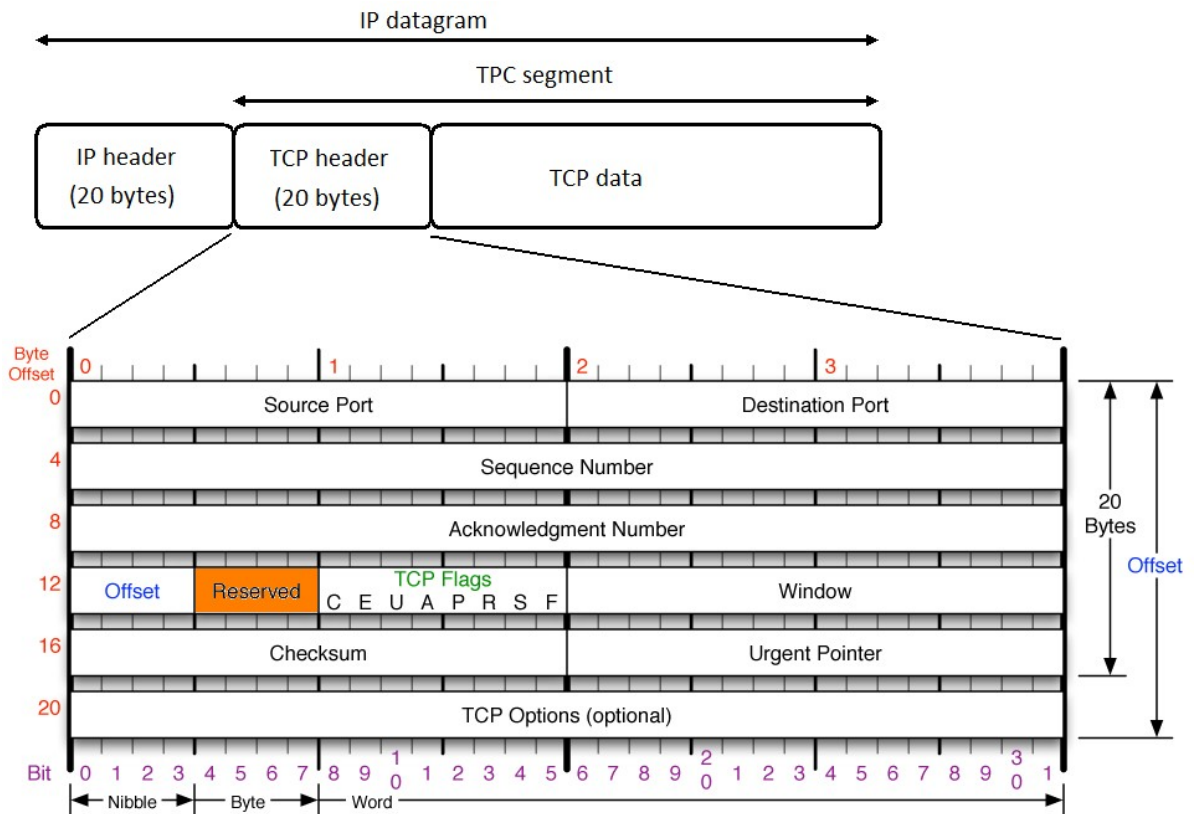
Nakonec je třeba zakódovat všechny tyto kvantované koeficienty do výsledného souboru. K tomu se v JPEG kompresi používá RLE, Variable Length Integer (VLI) kódování a Huffmanovo kódování. RLE umožňuje efektivněji uložit posloupnosti nulových koeficientů. Pomocí VLI jsou převedeny hodnoty na dvojice (Kat, Bity). Kat určuje kategorii danou jako číslo rozsahu, ve kterém se číslo nacházelo. Bity pak obsahují dodatečnou informaci o pozici, kde se v daném rozsahu číslo nachází. Pomocí statického Huffmanova kódování jsou pak kódována čísla Kat. Druhá složka (Bity) je kódována přímo, je přímo zapsána do výstupního souboru hned za Huffmanův kód symbolu odpovídajícímu kategorii Kat. Počet bitů, který se použije na její zakódování, je určen jednoznačně hodnotou Kat. Použije se vždy jedna Huffmanova tabulka pro jednu z barevných složek.⁹

Dekomprese grafického formátu JPEG spočívá v podstatě v inverzi ke kompresi. Nejdříve se provede dekódování Huffmanova kódu s načtením příslušného počtu bitů. Dvojice bitů se převede na odpovídající hodnotu kvantovaného koeficientu. Následně se provede vynásobení příslušnou konstantou k dosažení přibližné velikosti vztážené k původním koeficientům. Zde dochází ke kvantizační chybě, jelikož v cyklu komprese jsme obětovali určité procento kvality na úkor velikosti obrazové informace. Nyní rozdělíme data na 8x8 koeficientu, které podrobíme DCT v inverzním slova smyslu. Posledním krokem je převedení do požadovaného RGB barevného prostoru.

1.3.4.3 Metoda rezervních dat - TCP

Tato metoda není příliš rozšířená a v praxi se využívá spíše sporadicky. Její princip spočívá ve využívání dat, které jsou rezervována pro jiný účel. Například mohou sloužit pro budoucí potřeby, nebo naopak v dnešní době již ztratili smysl. Jako příklad lze uvést hlavičku protokolu TCP která je znázorněna na obrázku číslo 6.

⁹ ONDRUŠ, J. *Beztrátová komprese JPEG grafiky* [online]. 2009. 7 s



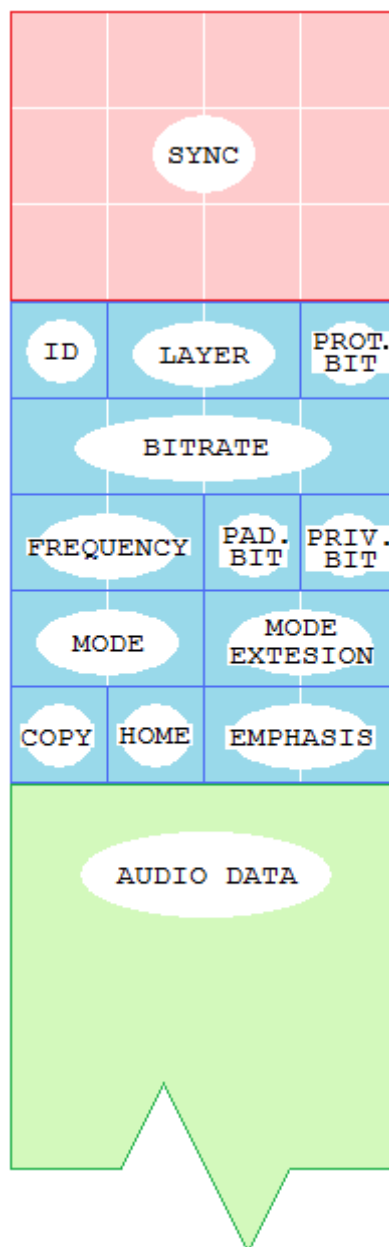
Obr. 6 – Hlavička protokolu TCP – dostupné z

<http://www.visi.com/~mjb/Drawings/>

Jedna hlavička protokolu TCP přenese 6 bitů. Význam rezervovaných dat není pro vlastní směrování podstatný a ani následné zpracování nebere na tyto data zřetel, proto jsou přímo předurčena pro jiné, v našem případě steganografické využití.

1.3.4.4 Metoda rezervních dat – MP3

Jako další příklad pro tuto metodu lze uvést zvukový formát mp3. Kodér tohoto formátu rozdělí zvukový signál na malé části, které se nazývají frames (rámce). Jedná se o data, které reprezentují zlomky vteřin audio signálu. Jejich velikost není konstantní a závisí na vzorkovací frekvenci, výsledném bitovém toku a samozřejmě na složitosti, rozmanitosti, dynamičnosti audio signálu. Jelikož je každý rámeček jinak veliký je zapotřebí rozlišit jeho začátek a konec. Každý rámeček tedy obsahuje tzv. hlavičku, což je sekvence 32 bitů (4 byty), vypovídající o základních parametrech audio signálu viz obrázek sedm. Tuto hlavičku obsahuje každý rámeček a jeho obsah je z velké části totožný.



Obr. 7 – *Hlavička rámce formátu MP3*

Jednotlivý význam bitů je přehledně uveden v tabulce číslo dva. Mnoho z nich není při zpracování souboru přehrávačem zapotřebí. Některé bity jsou červeně orámované, to jsou ty bity, do kterých je vhodné ukryt utajovanou zprávu. Hovoříme o uživatelském bitu, bitu autorské ochrany, bitu originality a bity zvýraznění.

Délka (bit)	Pozice (bit)	Popis informace																						
11	31-21	SYNC – synchronizace rámce (všechny bity nastaveny na hodnotu log. „1“)																						
2	20-19	ID – verze MPEG audia: 00 – MPEG verze 2.5 01 – Rezervováno 10 – MPEG verze 2 11 – MPEG verze 1																						
2	18-17	Layer – vrstvy 00 – Rezervováno 01 – Layer III 10 – Layer II 11 – Layer I																						
1	16	Protection bit 0 – Chráněno CRC (16 bitů CRC následuje hlavičku) 1 – Nechráněno																						
4	15-12	Bitrate																						
2	11-10	Sample frequency - vzorkovací frekvence																						
		<table border="1"> <thead> <tr> <th>Bity</th> <th>MPEG1</th> <th>MPEG2</th> <th>MPEG2.5</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>44100 Hz</td> <td>22050 Hz</td> <td>11025 Hz</td> </tr> <tr> <td>01</td> <td>48000 Hz</td> <td>24000 Hz</td> <td>12000 Hz</td> </tr> <tr> <td>10</td> <td>32000 Hz</td> <td>16000 Hz</td> <td>8000 Hz</td> </tr> <tr> <td>11</td> <td>Rezerv.</td> <td>Rezerv.</td> <td>Rezerv.</td> </tr> </tbody> </table>	Bity	MPEG1	MPEG2	MPEG2.5	00	44100 Hz	22050 Hz	11025 Hz	01	48000 Hz	24000 Hz	12000 Hz	10	32000 Hz	16000 Hz	8000 Hz	11	Rezerv.	Rezerv.	Rezerv.		
		Bity	MPEG1	MPEG2	MPEG2.5																			
		00	44100 Hz	22050 Hz	11025 Hz																			
		01	48000 Hz	24000 Hz	12000 Hz																			
10	32000 Hz	16000 Hz	8000 Hz																					
11	Rezerv.	Rezerv.	Rezerv.																					
00	44100 Hz	22050 Hz	11025 Hz																					
01	48000 Hz	24000 Hz	12000 Hz																					
10	32000 Hz	16000 Hz	8000 Hz																					
11	Rezerv.	Rezerv.	Rezerv.																					
1	9	Padding bit 0 – rámeček není „vycpán“ (všechny bity využity pro data) 1 – rámeček je „vycpán“ jedním oddílem																						
1	8	Private bit - uživatelský bit – volný (má pouze informační význam)																						
2	7-6	Mode - režim zpracování kanálů 00 – Stereo 01 – Joint stereo 10 – Dual channel (2 mono kanály) 11 – Single channel (mono)																						
2	5-4	Mode extension - Specifikace režimu zpracování kanálů																						
		<table border="1"> <thead> <tr> <th rowspan="2">bity</th> <th rowspan="2">layer I-II</th> <th colspan="2">layer III</th> </tr> <tr> <th>intens.stereo</th> <th>MS</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>4 pásma</td> <td>Off</td> <td>Off</td> </tr> <tr> <td>01</td> <td>8 pásem</td> <td>On</td> <td>Off</td> </tr> <tr> <td>10</td> <td>12 pásem</td> <td>Off</td> <td>On</td> </tr> <tr> <td>11</td> <td>16 pásem</td> <td>On</td> <td>On</td> </tr> </tbody> </table>	bity	layer I-II	layer III		intens.stereo	MS	00	4 pásma	Off	Off	01	8 pásem	On	Off	10	12 pásem	Off	On	11	16 pásem	On	On
		bity			layer I-II	layer III																		
			intens.stereo	MS																				
		00	4 pásma	Off	Off																			
01	8 pásem	On	Off																					
10	12 pásem	Off	On																					
11	16 pásem	On	On																					
00	4 pásma	Off	Off																					
01	8 pásem	On	Off																					
10	12 pásem	Off	On																					
11	16 pásem	On	On																					
1	3	Copyright 0 – Audio není chráněno autorskými právy 1 – Audio je chráněno autorskými právy																						
1	2	Home 0 – Kopie originálního média 1 – Originál																						
2	1-0	Emphasis – zvýraznění 00 – neplatná kombinace 01 – 50 / 15 ms 10 – Rezervováno 11 – CCIT J.17																						

Tab. 2 – Význam bitů v hlavičce rámce formátu MP3

Změna těchto bitů nemá vliv na integritu zvukového rámce. Nejhorší scénář, jež může nastat je, že v přehrávači nebude zobrazena informace o autorství nahrávky. Tím jsme tedy získali 5 bitů z každého rámce mp3 skladby, písnička délky přibližně 4 minuty obsahuje cca 8000 rámců, což znamená $8000 \times 5 = 45000$ bitů. Jeden znak ASCII kódu je reprezentován šesti bity, tzn. získá prostor pro přibližně 5000 znaků.⁸

1.3.5 Nejpoužívanější steganografická nosná media současnosti

Text – v tomto případě uvažujeme o krycím mediu jako o souboru obsahující čistý (holý) text tzn., že každý bit v daném souboru reprezentuje jeden znak, je jakákoliv steganografická metoda velice obtížně aplikovatelná. Změnou jakéhokoliv bitu změníme i primární informaci. Některé metody týkající se steganografie v textu jsem uvedl již v kapitole Historie steganografie (Cardanova mřížka, agenturní systém), jako další použitelné varianty bych uvedl techniky využívající sémantiku, syntaxi a formátování textu.

Právě techniky formátování, nám dávají asi nejvíce možností jak tajnou informaci ukrýt. Například metoda využívající prázdné místo v textu dokumentu. Vkládání zprávy spočívá v manipulaci s „bílymi“ místy v textu (např. mezery mezi znaky, mezi slovy, pozice počátečního nebo koncového znaku v řádku) a nepoužitého prostoru na stránce. Nejjednodušší je použít mezery mezi jednotlivými slovy – vložení binární zprávy bude realizováno jednou mezerou mezi slovy pro hodnotu „0“, nebo dvěma mezerami mezi slovy pro hodnotu „1“.¹⁰

Nicméně je zde jedna možnost, jak v dnešní době využít text jako nosné medium. SPAM neboli nevyžádané sdělení (pošta) je nečastěji doručeno na naši emailovou adresu, diskusní fóra nebo instant message (skype, icq). V dnešní době výzkumy uvádí, že spamy tvoří přes 80% veškeré komunikace realizované přes email. Tento globální nešvar zasáhl celou síť internet a je tedy velice lákavé využít ho jako náš „nevinný cover“. Nejjednodušeji jak tuto metodu využít a otestovat je na domovských stránkách, kde pomocí několika jednoduchých kroků vytvoříte spam obsahující vaše ukrytá data.

Zde je ukázka spamu, který byl vygenerován na výše uvedených stránkách.

⁸ MACIAK, L. *MP3 STEGONOGRAPHY* [online]. 2005. 17 s.

¹⁰ ZIGULEC, Š. *Steganografická komunikace* [online]. 2006. 24-25 s.

Dear Friend , Your email address has been submitted to us indicating your interest in our newsletter ! If you are not interested in our publications and wish to be removed from our lists, simply do NOT respond and ignore this mail ! This mail is being sent in compliance with Senate bill 2116 , Title 8 , Section 306 ! This is not multi-level marketing ! Why work for somebody else when you can become rich inside 78 months . Have you ever noticed nobody is getting any younger & people love convenience ! Well, now is your chance to capitalize on this ! We will help you increase customer response by 130% and use credit cards on your website ! You can begin at absolutely no cost to you ! But don't believe us ! Mr Jones of Missouri tried us and says "I've been poor and I've been rich - rich is better" . This offer is 100% legal ! We implore you - act now . Sign up a friend and you get half off . Thanks !

Dear Professional , Especially for you - this amazing announcement ! This is a one time mailing there is no need to request removal if you won't want any more ! This mail is being sent in compliance with Senate bill 1624 , Title 8 , Section 305 . This is not a get rich scheme ! Why work for somebody else when you can become rich in 55 WEEKS . Have you ever noticed nearly every commercial on television has a .com on in it & most everyone has a cellphone . Well, now is your chance to capitalize on this . We will help you deliver goods right to the customer's doorstep and process your orders within seconds ! The best thing about our system is that it is absolutely risk free for you ! But don't believe us . Mrs Jones who resides in Maryland tried us and says "Now I'm rich many more things are possible" . We are licensed to operate in all states . Because the Internet operates on "Internet time" you must hurry . Sign up a friend and you get half off ! Cheers !

Pro dekódování tohoto spamu je zapotřebí zadat heslo: *lomikar*, výsledná utajená zpráva má tvar: *plan was disclosed*.

Na konec této kapitoly, bych rád uvedl veselý příklad steganografie. Znamý moderátor BBC, James May, účinkující v pořadu Top Gear, prostřednictvím svých recenzí motoristickém týdeníku publikoval svůj vyhraněný názor, týkající se těžké práce redaktora. Utajená zpráva byla skryta ve velkých červených písmenech, jimiž začínala každá recenze. Když na to vedení britského magazínu Autocar přišlo, James May byl propuštěn. Vše je přehledně vidět na obrázku číslo osm.



Obr. 8 – James May, Road Test Yearbook Issue (1992)-dostupný z

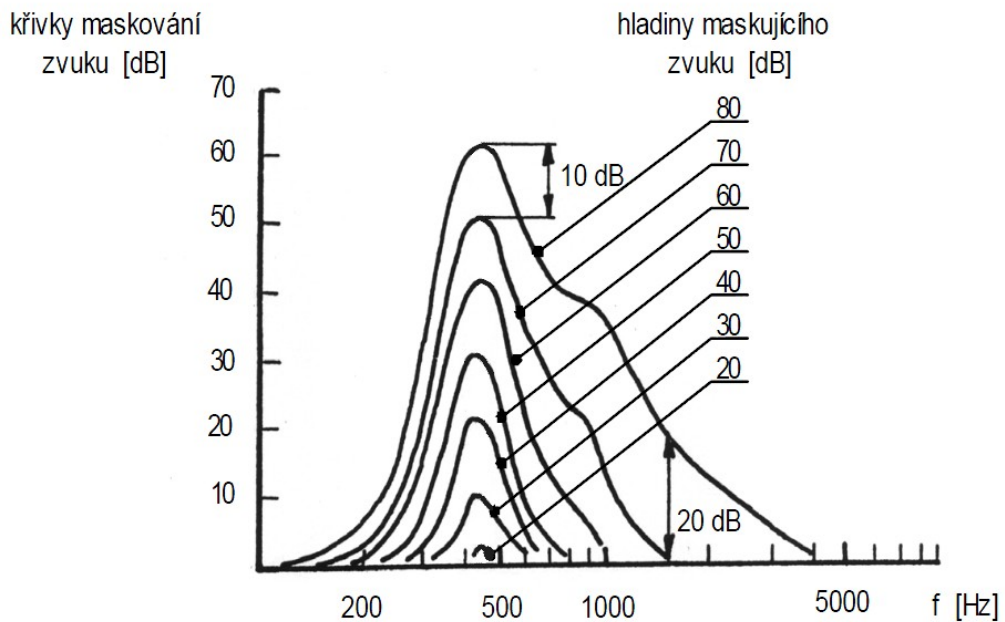
http://en.wikipedia.org/wiki/James_May

Obrázky – asi nejvíce rozšířené nosné medium, které se využívá ve steganografii. Mezi jeho nesporné výhody lze zařadit obrovskou rozšířenost. Internet je doslova nappán obrázky různých formátů, velikostí i tématik. Proto nevzbuzuje mezi uživateli sebemenší podezření z možného obsahu utajované informace. Navíc je obraz vnímán méně dokonalým smyslem nežli například zvuk. Lidský zrak není těžké ošálit, a jeho citlivost je v porovnání se sluchem malá. Proto si nevšimneme malých diferencí odstínů, sytosti a jasů, které využíváme pro ukrytí tajné zprávy. Nejčastěji využívaná metoda je zde změna bitů na pozici LSB.

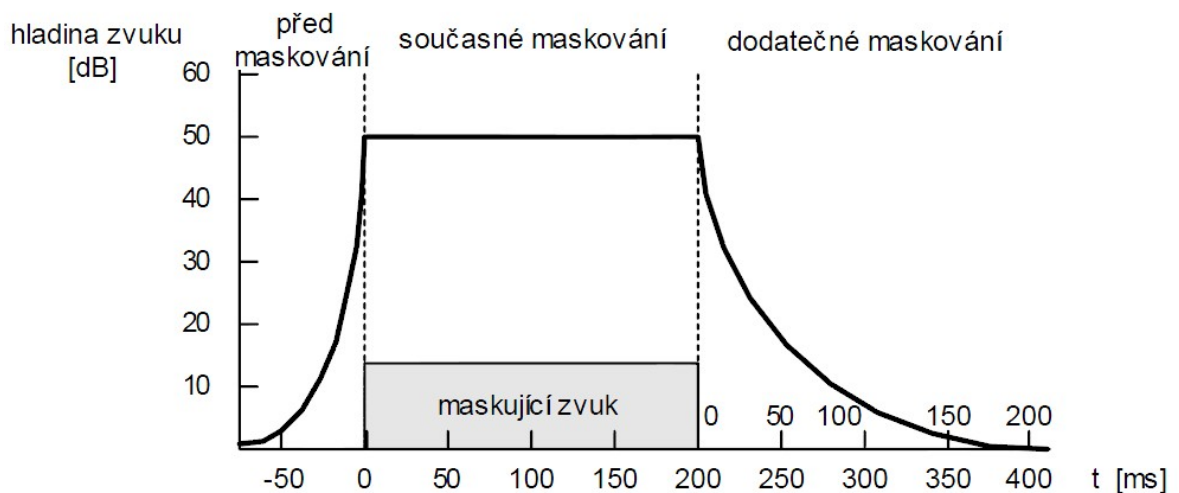
Poslední metoda, která stojí za zmínku, se vyznačuje svojí robustností. Nazývá se Patchwork (sešívání) a je založena na pseudonáhodném výběru dvojice bitů, které jsou následně pozměněny (zesvětlení, ztmavení). Pomocí tohoto principu je zpráva opakovaně ukryvána do celého krycího media. Tím je zaručeno dekódování informace i v případě, kdy byl obrázek převeden pomocí ztrátové komprese do formátu JPEG se zachováním kvality 75 %. Robustnost této metody dovoluje značné úpravy obrázku jako například oříznutí, zvětšení, aplikace filtrů i již zmiňovaná změna formátu a přesto je zde vysoká pravděpodobnost, že skrytá zpráva nebude porušena.

Audio záznam – lidský sluchový orgán je co do frekvenčního (15 Hz – 20 kHz) tak i dynamického rozsahu (2×10^{-5} Pa – 10^2 Pa) velice flexibilní. Ovšem i zde jsou jisté hranice, které jsou pro náš sluch limitující. Tak zcela určitě můžeme utajovanou zprávu ukryt do frekvenčního pásma sahající nad 20 kHz. Jednoduchým namodulováním na nosnou frekvenci (např. 30 kHz) se dostaneme do neslyšitelné oblasti. Značná nevýhoda této metody spočívá v nadměrném nárůstu velikosti zvukového souboru.

Druhá metoda je dána fyziologickými vlastnostmi ucha. Jak jsem popisoval ve své bakalářské práci, jedná se o „maskování“. Tedy případ kdy do sluchovodu dorazí dva signály naráz. Výsledkem takového děje může být zastření nebo úplné utlumení jednoho ze signálů. Na maskování se podílí nejen intenzita zvuku, ale i frekvence přijatých signálů jak je vidět na obrázcích. Maskování probíhá nejen ve frekvenční oblasti ale i v časové ose. Je tedy velice výhodné ukryt data před posluchačem právě do míst, kde se maskovací efekt projevuje. Obrázek číslo devět zobrazuje, jak maskovací efekt ovlivňuje hlasitost, zatímco desátý obrázek ukazuje maskování v časové oblasti.



Obr. 9 - Maskovací audiogram. Maskující zvuk je šum o šíři pásma 90 Hz a středním kmitočtu 450 Hz s hladinami 20-80 dB, dle F. Kadlece⁵



Obr. 10 - Časové oblasti současného maskování, před-maskování a dodatečného maskování zvuku, dle F. Kadlece⁵

Obecně lze prohlásit, že metodu zápisu dat na pozici LSB (u audio-coveru se častěji uvádí zkratka LBE – low bit encoding, což je analogie k LSB) můžeme použít u všech digitálních medií (nosičů), které jsou reprezentovány binárním vyjádřením a jejichž data lze měnit. Tedy i audio signál je vhodný adept na tuto steganografickou metodu. Shodně jako u grafických formátů se zde ukrytá informace projeví formou šumu.

Další z metod hojně využívaných u audio-coveru je metoda fázového kódování. Data jsou kódována substitucí fáze segmentu audio signálu pomocí referenční fáze, která reprezentuje ukryvaná data. Tento postup vnáší do audiosignálu fázovou disperzi a nespojitosti fázového průběhu závislé na každém frekvenčním segmentu. Minimalizace fázové disperze je závislá na použité přenosové rychlosti.¹⁰

Metoda, která rozprostře utajenou informaci, do co nejširší frekvenční oblasti daného zvukového záznamu se nazývá, metoda rozprostřeného spektra (spread spectrum). Výhoda této metody spočívá v obtížnějším detekování, obzvláště je-li před samotným ukrytím zpráva zašifrována. Posluchač může ve stegogramu reflektovat skrytá data jako šum.

Video – do tohoto druhu média, můžeme uložit tajná data hned několika způsoby. Máme-li na mysli video, mluvíme o kombinaci obrazové a zvukové informace, která je pozorovateli prezentována současně. Proto můžeme zvolit jak metody využívané pro audio-cover, tak metody využívané u obrazové informace. Především hovoříme o LSB, maskování v časové ose či frekvenční maskování.

Datagramy komunikačních protokolů – řada protokolů rezervuje ve svých diagramech sekce, které samy nepoužívají či používají jen výjimečně. Nepoužívané sekce mohou existovat jako záloha pro případné budoucí využití (například rozšíření protokolu o další funkce nebo zvětšení bitové šířky některých jiných sekcí) nebo se jejich prvotní účel naopak stal časem bezpředmětný. Takovéto oblasti mohou být využity pro embedding tajných informací. Příjmová strana musí mít schopnost sledovat síťovou komunikaci na té vrstvě, jejíž datagramy byly pro přenos použity.⁴

Další možností jak ukrýt tajnou zprávu do tohoto druhu media je využít takové datagramy, které umožňují připojit dodatečná data. (např. data vyšší síťové vrstvy). Jedná se o datagramy používané v asynchronní komunikaci.

Spustitelné binární soubory – spustitelný binární soubor není nic jiného, než posloupnost instrukcí. Instrukce samozřejmě nelze měnit, tím bychom zcela změnili chod programu. Využijeme-li však instrukce nepodmíněného skoku a tím vytvoříme části souboru, které

¹⁰ ZIGULEC, Š. *Steganografická komunikace* [online]. 2006. 27 s.

⁴ ŽILKA, R. *Steganografie a stegoanalýza* [online]. 2008. Diplomová práce. 14 s.

nikdy nebudou provedeny. Do nich pak ukryjeme zprávu. Tento způsob se sice poměrně snadno detekuje, ale kdo z nás běžně hledá ve spustitelných souborech hluchá místa? Jinou metodou je ukryvání do nepoužitých proměnných, pomocí negativní / pozitivní formulace podmínek skoku, atd.⁶

Steganografický souborový systém – jedná se o systém, který dokáže alokovat nevyužité místo na pevných a flash discích. Toto volné místo může využít pro zápis tajných informací. Je zde však velké nebezpečí přerušení integrity dat či přemazání, v případě, že na volné místo disku uložíme nějaká jiná data. Nejznámější steganografický souborový systém pro Linux je StegFS.

1.4 Stegoanalýza

Jak již bylo zmíněno na začátku diplomové práce je stegoanalýza disciplína, jenž má za úkol detekovat, extrahovat, degradovat a popřípadě pozměnit skrytou informaci v krycím mediu.

Jak v kryptografii tak i ve steganografii jsou „luštitelé a analytici“ v nelehkém postavení, mají-li přijít na metody a principy jak získat skrytý obsah komunikace, oproti jejich kolegům, jež navrhují rafinované metody ukrytí tajných dat. Dalo by se dokonce říci, že stegoanalytici vyčkávají na svou kapku štěstí, aby odhalili nebo uhádli utajenou informaci. Jejich práce je náročná právě proto, že běžně vyhlízející všednosti mohou skrývat cenná tajemství. Peter Wayner ve své knize *Disappearing Cryptography - A book on steganography, information hiding, watermarking and other techniques for disguising informatik* rozdělil základní způsoby detekce steganografie na tyto tři:

Visual or aural (vizuální a sluchová) – při této detekci je předpoklad použití svých vlastních smyslů. Především se jedná o stegogramy, u kterých není očekávána nějaká analýza lidskými smysly. Naše oko ale hlavně náš sluch, dokáže rozpoznat i celkem malé difference, nesrovnalosti, obzvlášť v případě, že krycí cover je nám známý nebo máme-li ho k porovnání. Při pohledu na oba soubory pak můžeme nabýt dojmu určitého pozměnění. Bezpochyby se však jedná, o subjektivní způsob detekce.

⁶ FOLTÝNEK, T. *Komprimace a šifrování* [online]. 2010. 82-83 s.

Structural (strukturální) – tato metoda je na rozdíl od prvně jmenované jednoznačně strojového charakteru. Krycí medium je zde podrobena analýze, která by měla na jakékoliv nestandardní a nelogická struktury poukázat. Právě v těchto místech může být prostor pro ukrytí tajné zprávy. Jedná se například o metody formátování, kdy je v textu neúměrný nadbytek volných řádků a bílých mezer.

Statistical (statistická) – 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.¹¹

Daleko lépe lze použít statistickou metodu v případě, máme-li k dispozici stegogram i čistý krycí medium daného typu. Nyní je zapotřebí zjistit co nejvíce charakteristik týkající se krycího media. Ty použijeme při sestavování analyzátoru, který se bude snažit identifikovat podezřelé soubory (stegogramy) na základě nastřádaných informací. Tento model využil ve své diplomové práci Ing. Jiří Hološka, který pomocí neuronových sítí zhotovil stegoanalyzátor, který s velkou úspěšností detekoval stegogramy ve velkém množství obrazových souborů.

Analytik Neil Johnson sestavil tabulku, která nám názorně ukazuje šest detekčních metod stegoanalýzy v závislosti na dostupném množství informací jak ukazuje tabulka číslo tři.

	<i>stegogram</i>	<i>original cover</i>	<i>hidden message</i>	<i>steganography algorithm</i>
<i>stego only</i>	•			
<i>know cover</i>	•	•		
<i>know message</i>	•		•	
<i>chosen steganography</i>	•			•
<i>chosen message</i>	•			
<i>know steganography</i>	•	•		•

Tab. 3 – *Metody steganografie*

¹¹ THAMPI, SABU M. *Information Hiding Techniques: A Tutorial Review* [online]. 2004. 14 s.

Metoda stego-only značí, že při stegoanalýze pracujeme pouze s jedinou informací a to s vlastním stegogramem. Jedná se tedy o jednu z nejtěžších metod, která nás může dovést k odhalení utajené informace. Proto se využívá převážně k pouhému zjištění, zdali analyzovaný soubor nese charakteristiky poukazující na přítomnost embeddingu.

Při know-cover metodě se přímo nabízí porovnávat nejrůznějšími metodami stegogram a originální medium, které máme k dispozici a tím zjistit změny, jenž můžeme zobecnit pro statistickou stegoanalýzu nebo které nám přímo odhalí ukrývanou informaci. V JPEG mediu by skrytá zpráva mohla být reprezentována šumem, a pokud by nám šlo o pouhé zničení utajených informací při použití metody LSB, není nic jednoduššího než všechny bity na této pozici nastavit do hodnoty logické nuly.

Postup nazvaný know-message poskytuje řešiteli znalost utajené informace, čímž můžeme zjistit podrobnosti týkající se použitého stegosystému.

Speciální kategorie chosen-message zahrnuje experimentální praktiky, které se opakovaným ukrýváním různých zpráv pomocí různých stegosystémů (a následných hledání společných znaků s primárním zkoumaným stegogramem) snaží zjistit bližší detaily jako např. potenciální použitý stegosystém. Tyto metody však neumějí vyvrátit stopy steganografie. Postupy typu chosen-steganography realizují podobný experimentální proces, avšak přítomnost steganografie je apriori potvrzena, včetně udání stegosystému.⁴



Obr. 11 - *Stegoanalýza* dostupný: <http://www.jjtc.com/Steganalysis/>

⁴ ŽILKA, R. *Steganografie a stegoanalýza* [online]. 2008. 38 s.

Jinak se na problematiku dělení steganografie z pohledu stegoanalýzy dívají odborníci Neil F. Johnson a Sushil Jajodia v článku „Steganalysis of Images Created Using Current Steganography Software“. Ti zavádějí dva termíny a to *image domain* a *transform domain*.

Nástroje a metody *image domain* pracují na bitové bázi. Tyto postupy jsou nazývány také „simple systems“ - „jednoduché systémy.“ Jedná se o aplikaci šumu a změnu nejméně významných bitů souborů. Mezi konkrétní nástroje této kategorie patří například: StegoDos, S-Tools, Mandelsteg, EzStego, Hide and Seek (verze 4.1), Hide4PGP, Jpeg-Jsteg, White Noise Storm, and Steganos. Formáty dat využívané těmito metodami jsou bezztrátové a data mohou být vyzvednuta a může být s nimi manipulováno.¹²

Metody „transform domain“ zahrnuje přímé transformace dat. Jedná se například o změnu luminiscence. Nástroji jsou například PictureMarc, JK-PGS, SysCop, nebo SureSign. Tyto metody jsou obvykle mnohem robustnější než předchozí typ a často také nezávislé na druhu média. Mohou tak „přežít“ například konverzi obrázku z bezztrátového na ztrátový formát.¹²

Každá metoda steganografie ukrývá požadovanou informaci velice různorodě. Proto je získání informace, pokud útočník nezná použitou metodu, velice komplikovaný proces. „Naštěstí“ některé steganografické nástroje generují určité charakteristiky, dle kterých se dá použitý nástroj rozeznat.¹²

Základním postupem stegoanalýzy je porovnání originálního obrazu s obrazem, nesoucím kódové slovo. Jedná se vždy o jemné odlišnosti, které by pouhým pozorováním, bez porovnání, unikly pozornosti útočníka. Originální obraz je tedy velice přínosná pomoc při získávání klíčových slov.¹²

¹² BĚLONOHÝ, R., JEŽEK, a kol. *Steganografie 2* [online]. 2003.

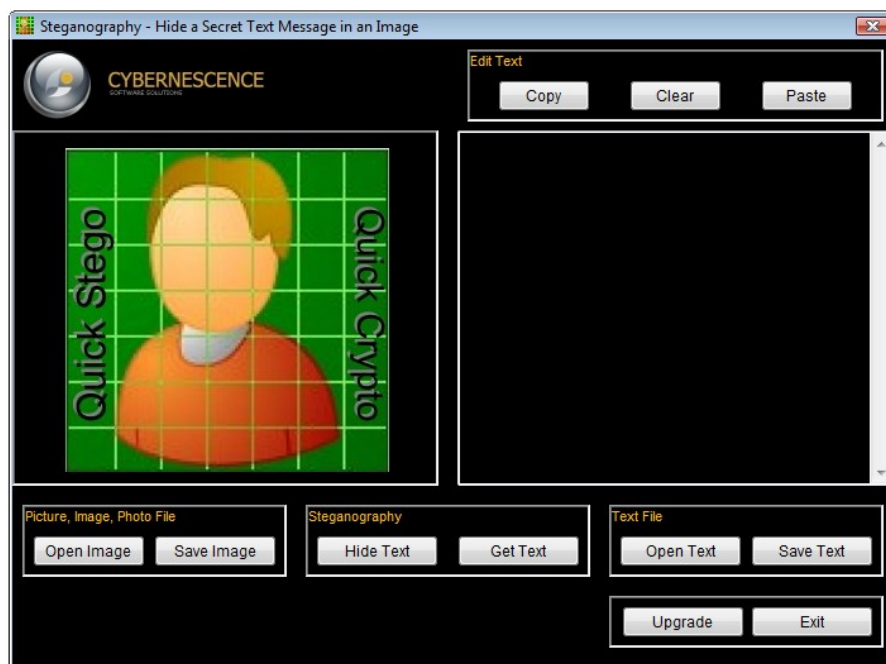
II. PRAKTICKÁ ČÁST

2 STEGANOGRAFICKÉ NÁSTROJE

Cílem práce je popis několika steganografických aplikací, které následně testuji na množině deseti grafických medií s proměnou velikostí. Do těchto krycích souboru je vloženo několik typů ukryvaných dat, o různé velikosti a struktuře. Výsledná data jsou podrobena analýze ve smyslu měření několika charakteristických veličin. Míru změny lze odečíst z jednotlivých hodnot nebo z přiložených grafů. Na závěr je vyhodnocena nejvhodnější steganografická aplikace s odůvodněním pro daný krycí cover.

Zde je popsáno několik steganografických softwarů, které svými specifickými vlastnostmi, jednoduchostí a efektivností zasluhují pozornost.

Jako první je uvedena webová aplikace Spammimic, která během okamžiku vytvoří SPAM, jenž obsahuje Vámi vloženou ukrytou zprávu. Jedná o perfektní variantu utajené komunikace. Program Quick Stego nabízí jako většina zde uvedených steganografických systémů vložení tajných dat do obrázku ve formátu BMP, JPEG a GIF s tím rozdílem, že můžeme vložit pouze čistý text. Ten je psán přímo do okna k tomu určené (viz obrázek dvanáct). Jedná se tedy o velice jednoduchou přesto efektivní a rychlou variantu, jak utajit komunikaci. Ovládání je velice intuitivní, například vyvolání skrytého textu je provedeno automaticky, po načtení daného stegogramu.



Obr. 12 - – Uživatelské rozhraní programu *Quick Stego*

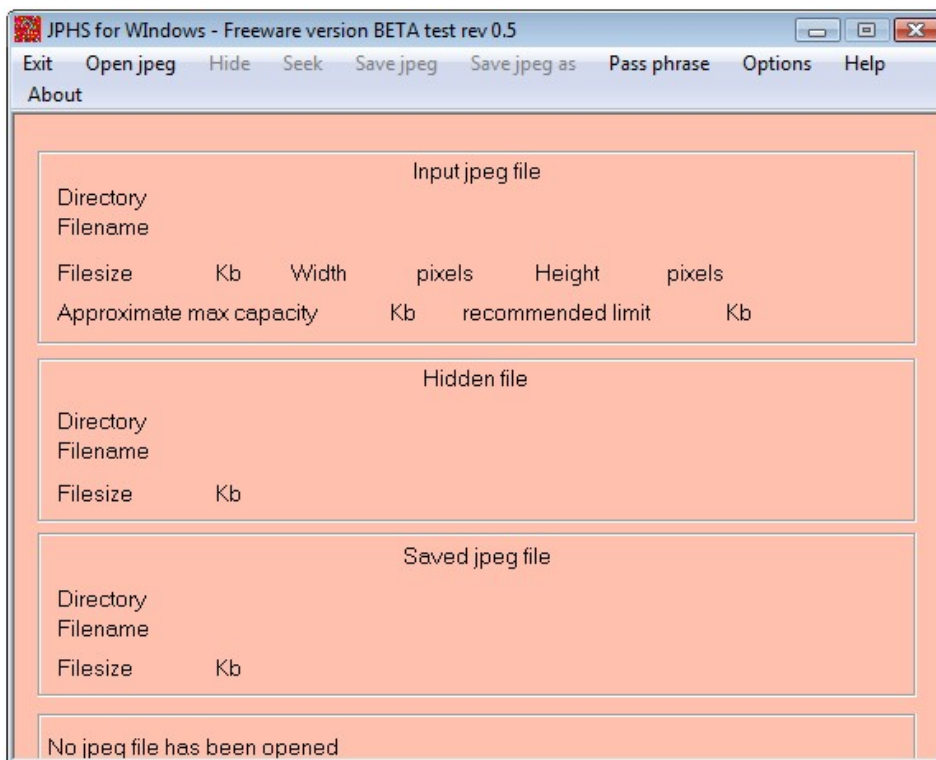
Na internetu je mnoho steganografických programů, které nabízejí doplňující funkce jako kryptografickou ochranu vybraných sekcí na disku, správce hesel, šifrování e-mailů i souborové manažery. Tyto nadstandardní funkce a propracovaná grafická rozhraní ospravedlňují jejich zpoplatnění. Jako případného zástupce lze uvést aplikaci Steganos či Invisible secret.

2.1 JP Hide and Seek

První program, použitý v diplomové práci je JP Hide and Seek (též JPEG Hide and Seek), a byl vytvořen Allanem Lathamem v prostředí Visual C++ v roce 1998. Pro svou jednoduchost a freewarovou distribuci, je k sehnání i dnes.

Pro práci s programem není potřeba jeho instalace pouze rozbalení. Má jednoduché uživatelské rozhraní (to je vyobrazeno na obrázku číslo třináct), což výrazným způsobem zefektivňuje a urychluje práci. Steganografická kapacita je nastavena v poměru 5 % k velikosti originálního obrázku. Tím je zaručena naprostá věrohodnost a obrázek nebude vzbuzovat podezření.

Samotné ovládání je velice intuitivní a jednoduché. Po spuštění je zapotřebí odsouhlasení podmínek, po němž se objeví uživatelské prostředí. Pomocí několika kroků (načtení krycího JPEG souboru, vložení hesla pro extrahování utajených informací, načtení jakéhokoli souboru pro ukrytí splňující podmínku velikosti, umístění nového JPEG souboru a jeho případně přejmenování), lze vytvořit přesně definovaný stegogram. V případě, že ukryvaný soubor je příliš velký zobrazí se chybová hláška. Aby k tomuto jevu nedocházelo příliš často, je zde ukazatel, který vám vypočítá maximální a doporučenou velikost ukryvaného souboru z krycího JPEGu.



Obr. 13 – Uživatelské rozhraní programu JPHS

2.2 Steghide

Program vytvořil Stefanem Hetzlem a jeho nejaktuálnější verze je z roku 2003. Tato verze Steghide 0.5.1 byla použita pro embedding. Jedná se o steganografický program, který nabízí uživateli ukrytí tajné informace do několika krycích medií. A to WAV, AU jako zvukové covery a JPEG, BMP jako obrazové covery. Tento software využívá metodu vkládání dat, která minimalizuje pravděpodobnost odhalení při použití frekvenční analýzy. Celá metoda spočívá v proceduře na krycí medium, při které se zjistí maximální možná steganografická kapacita tak, aby výsledný stegogram byl z pohledu frekvenční analýzy shodný s frekvenční analýzou krycího coveru. Na oficiálních stránkách se lze dále dočíst:

- Kompresi vkládaných dat;
- šifrování vkládaných dat;
- využití kontrolního součtu k ověření integrity extrahovaných dat.

Používání programu není zpoplatněno, jelikož je distribuován jako GNU General Public License (GPL).

Program nemá grafické uživatelské rozhraní. Ovládá se pomocí příkazového řádku. Na webových stránkách je podrobný návod.

2.2.1 Steghide v příkazovém řádku ¹³

Vkládání dat je reprezentováno touto základní syntaxí:

```
$ steghide embed -cf vstupniobraz.jpg -ef zprava.txt
```

Enter passphrase:

Re-Enter passphrase:

```
embedding " zprava.txt " in " vstupniobraz.jpg "... done
```

Rozšířená syntaxe

```
steghide embed -p heslo -cf vstupniobraz.jpg -ef zprava.txt -sf vystupniobraz.jpg
```

-p přepínač pro vložení hesla.

-cf definuje vstupní soubor, do kterého budou vkládána data.

-ef definuje soubor se vkládanou zprávou.

-sf definuje výstupní soubor, pokud tento přepínač není použit, pro výstup se použije vstupní soubor.

Další parametry, které lze použít

-Z vkládaná data nebudou komprimována.

-K do výstupního obrazu se nebude vkládat CRC32 kontrolní součet, tento přepínač lze použít, pokud zpráva již obsahuje nějaký druh kontrolního součtu nebo pokud potřebujete uvolněných 32 bitů použít pro vložení zprávy.

-N do výstupního obrazu se nebude ukládat jméno souboru se vkládanou zprávou, při použití tohoto přepínače bude nutné pro extrahování zprávy určit jméno, do kterého se uloží zpráva.

Vyjmutí či získání tajné informace se provádí následně:

```
$ steghide extract -sf vystupniobraz.jpg
```

Enter passphrase:

¹³ HOLOŠKA, J. *Odhalování steganografie pomocí neuronových sítí* [online]. 2008. Diplomová práce. 33s.

wrote extracted data to "zprava.txt "

-sf definuje soubor obsahující skrytou informaci

Další parametry, které lze použít:

-xf definuje jméno výstupního souboru, tento přepínač je nutné použít, pokud se při vkládání zamezilo vložení jména souboru se zprávou.¹³

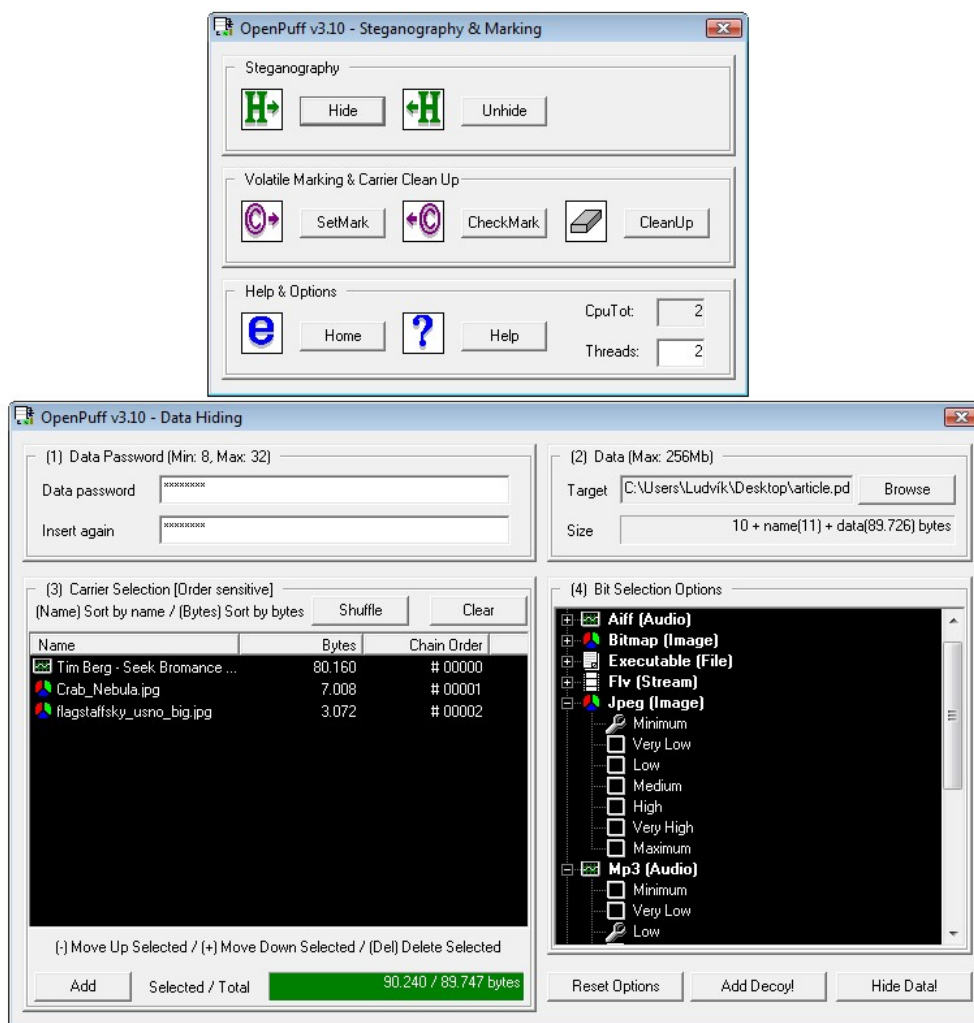
2.3 OpenPuff v3.10 Steganography & Watermarking

Program lze využívat bezplatně. Autor programu Cosimo Oliboni hovoří o jedinečném softwaru, který nabízí unikátní možnosti, jež v jiných steganografických systémech nenalezneme.

Program není potřeba instalovat, pro spuštění stačí jeho rozbalení. Software disponuje jednoduchým a přehledným grafickým uživatelským rozhraním (viz obrázek 14), které zefektivňuje a usnadňuje práci. Výhody tohoto programu jsou velká podpora krycích medií (grafické formáty BMP, JPG, PCX, PNG, TGA - zvukové formáty AIFF, MP3, NEXT / SUN, WAV - video formáty 3GP, MP4, MPG, VOB - ostatní FLV, SWF, PDF), možnost uložení až 256 Mb tajných informací, využití generátoru pseudonáhodných čísel (CSPRNG - cryptographically secure pseudo-random number generator) pro 256 bitovém vytváření symetrického klíče a uživatelské nastavení bitové hladiny LSB.

Mezi funkce programu je řazena možnost vložení tajné informace nejen do jednoho krycího media, ale do několika rozdílných souborů najednou. Tím je utajená informace rozprostřena a výsledný soubor je téměř nepozměněn. Pro srozumitelnost uvádím příklad: ukryt audio nahrávku o velikosti 1 Mb u ostatních steganografických systému vyžaduje nalezení jediného krycí coveru odpovídající velikosti (řádově desítky Mb). OpenPuff ale umožňuje uložení této nahrávky do několika souboru např. 2x PDF, 4x JPEG, 1xDOC, 1xMP3. Z logiky věci tedy vychází, že jednotlivé covery se změní minimálně. Pro extrahování tajné informace jsou zapotřebí všechna media, která se podílela na ukrytí.

¹³ HOLOŠKA, J. *Odhalování steganografie pomocí neuronových sítí* [online]. 2008. Diplomová práce. 33-34 s.



Obr. 14 – Uživatelské rozhraní programu OpenPuff

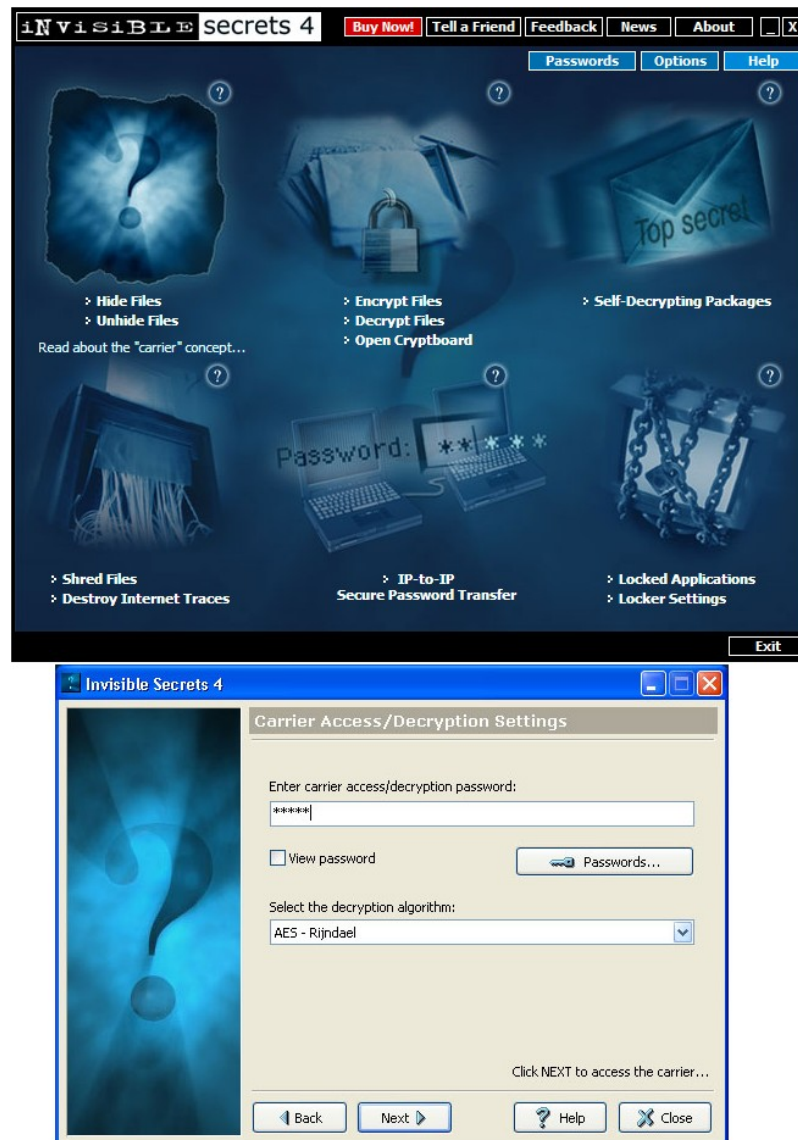
Podrobný návod v anglickém jazyce jak ovládat program OpenPuff je na oficiálních stránkách.

2.4 Invisible secrets 4

Jde o první zpoplatněný software použitý v diplomové práci. Rumunská společnost, která stojí za tímto programem, jej nabízí od 40 dolarů za licenci na jeden počítač. V případě, že si nejste jisti koupí, máte patnáctidenní zkušební dobu na otestování všech možností a funkcí. Nejedná se již pouze o čistě steganografický systém, ale tento program nabízí i možnost zabezpečit vaše soukromé složky v počítači pomocí kryptografických metod s využitím nejmodernějších šifrovacích algoritmů (AES - Rijndael, Twofish, RC 4, Cast128, GOST, Diamond 2, Sapphire II, Blowfish), nebo bezpečně smazat data tak, aby jejich obnova již nebyla možná. Naleznete zde i velice populární správce hesel, který si pamatuje všechny hesla a při jejich vytváření využívá pseudo-náhodný generátor. Další

funkce jsou IP-to-IP password transfer, který umožňuje bezpečný převod hesel mezi dvěma počítači pomocí šifrované internetové linky. Co se týká steganografických funkcí, program nabízí obligátní ukrytí do grafických a zvukových souborů. Mezi krycí media patří JPEG, PNG, BMP, HTML (ukrytí informace do webové stránky) a WAV. Samozřejmostí je ochrana skrytých dat heslem.

Invisible secrets 4 disponuje graficky propracovaným uživatelským prostředím což je zřejmé na první pohled z obrázku 15. V tomto případě program disponuje pouze anglickou lokalizací, nicméně ovládání je intuitivní a velice jednoduché. Za efektivní považuji vnoření programu a jeho funkcí do nabídky, kterou vyvoláme standardním kliknutím pravým tlačítkem na daný soubor. Další pozitivní funkci shledávám v implementaci nápovědy do menu tak, jak jsme zvyklí z ostatních komerčních programů. Pro použití program vyžaduje instalaci. Více informací lze získat z oficiálních stránek.



Obr. 15 - Uživatelské rozhraní programu Invisible secrets 4

2.5 Steganos

S německým programem Steganos je možné se setkat na trhu již od roku 1996. Tento konkrétní je z roku 2010 a jeho celý název zní Steganos Privacy Suite 12. Jedná se o placený software, který je možno vyzkoušet po dobu 30 dní zdarma. Nadále je požadováno zakoupení licence. Aplikace neobsahuje pouze steganografický systém, který ukládá tajná data do zvukového či obrazového (v našem případě JPEG) coveru, ale je zde mnoho užitečných funkcí jako např. Safe, což je šifrovací utilita, který bezpečně chrání vybraná data před neoprávněným uživatelem. Další nabízené funkce jsou, Portable safe – bezpečné šifrování na přenosných discích (CD, DVD, Flashdisk ...), Password manager – správce hesel, Private favorites – správa oblíbených internetových stránek i s hesly, E-mail

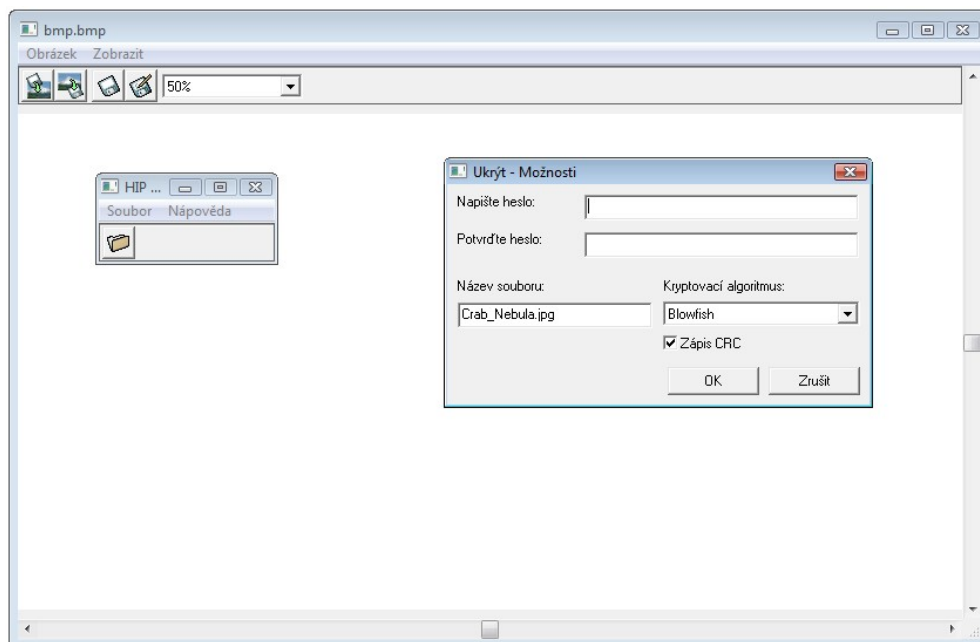
encryption – šifrování emailové pošty, TraceDestructor – odstranění stop při práci s počítačem a Shredder – mazání složek a souboru bez možnosti jakékoliv obnovy. Program využívá při šifrování 256 bitový AES algoritmus. Jelikož je tento program zpoplatněný očekává se odpovídající grafické uživatelské rozhraní (obrázek šestnáct), systémová stabilita, bezproblémové ovládání.



Obr. 16 - Uživatelské rozhraní programu Steganos

2.6 Hide In Picture

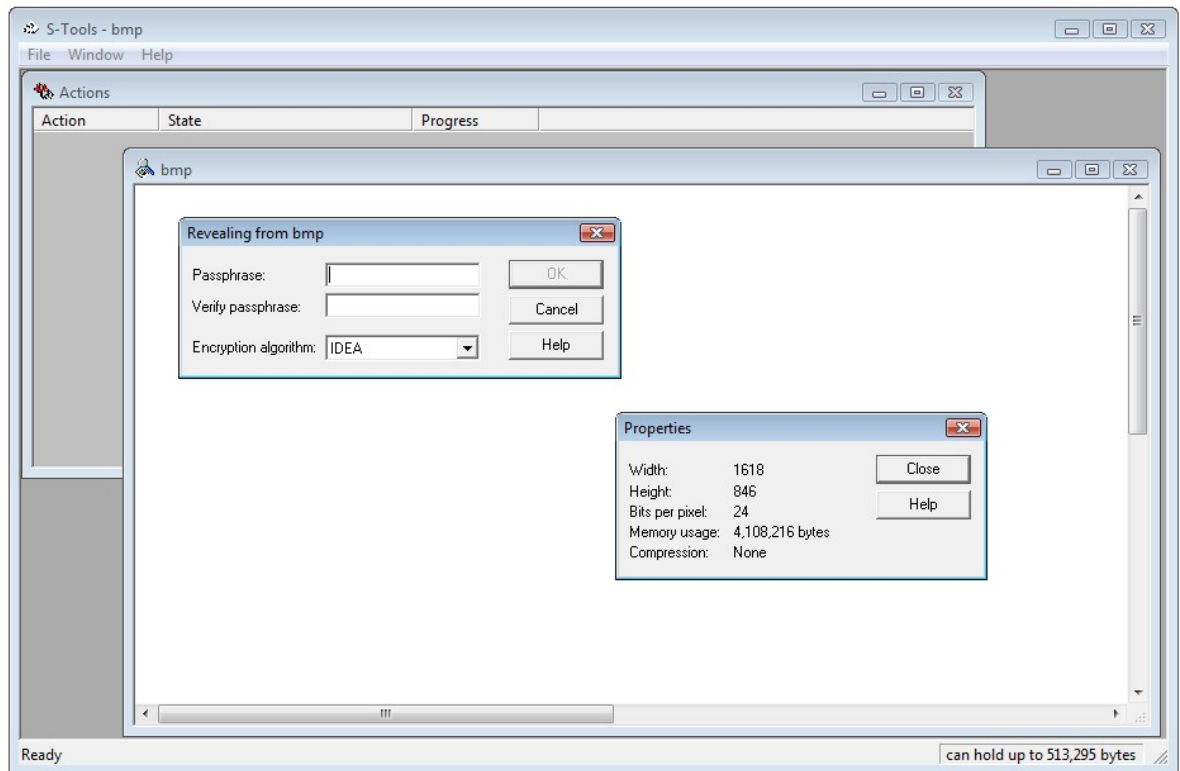
Tento program byl vytvořen Davidem Figueiredo. Jedná se o steganografický systém, jež skrývá tajná data do obrazové informace. Podporovány jsou dva formáty a to BMP a GIF. Verze, kterou používám ve své práci je nejnovější (HIP 2.1) a byla zveřejněna roku 2002. Software disponuje velice jednoduchým grafickým rozhraním, jak je vidět na přiloženém obrázku 17. Ukrytá data jsou chráněna heslem ve formátu symetrické blokové šifry AES - Rijndael nebo blowfish. Navíc je zde možnost vložení kontrolního součtu CRC. Metoda vkládání utajených informací je založena na změně bitů na pozici LSB. Program není potřeba instalovat a je volně šiřitelný pod licenčními podmínkami GNU General Public License (GPL). Stáhnou jej lze na domovské stránce.



Obr. 17 - Uživatelské rozhraní programu Hide In Picture 2.1

2.7 S – Tools

Andy Brown, tvůrce tohoto steganografického programu, jej zveřejnil v roce 1996 a to již ve čtvrté verzi 4.00. Ta je volně šiřitelná a lze ji stáhnout z některých internetových stránek (viz. použité zdroje). Domovské stránky jsou v dnešní době již nedostupné. Software nabízí ukládání skrývaných informací do obrazového media ve formátu BMP a GIF a audio coveru, ten musí být reprezentován WAV formátem. Jednoduché grafické rozhraní vyobrazeno na obrázku číslo osmnáct, které neobsahuje téměř žádné tlačítko, působí stroze. Ovládání je netradiční, a spočívá v přetahování jednotlivých souborů přímo do okna programu. K provedení jakékoliv akce je použito pravé tlačítko myši. Při vkládání skrývaných dat je zapotřebí vložit heslo, zde je na výběr mezi několika šifrovacími algoritmy a to IDEA, DES, TripleDES a MDC. Program není nutno instalovat.



Obr. 18 - Uživatelské rozhraní programu S - Tools

3 TESTOVÁNÍ

Testování výše popsaných steganografických programů proběhlo ve dvou samostatných fázích. Nejdříve byla ukryta data do obrazového formátu JPEG, který patří mezi nejrozšířenější vůbec. Jeho velkou výhodou je zachování kvality v poměru k velikosti. Tento nezpochybnitelný fakt stojí za tím, že skoro všechny obrázky na webových stránkách, šířených Internetem, jsou právě v tomto formátu. Ke skrytí dat bylo použito čtyř steganografických nástrojů a to, JP Hide and Seek, Steghide, Steganos a Invisible secrets 4.

Druhou samostatnou částí bylo využití obrazového formátu BMP (Windows Bitmap) pro ukrytí tajných dat. Jedná se o grafickou informaci reprezentovanou rastrovou grafikou. Tento formát není již v dnešní době hojně používán, ale jeho podporu garantuje drtivá většina grafických prohlížečů. Právě kvůli extrémní jednoduchosti a jeho masivní podpoře je mnoho steganografických systémů, jenž nabízí embedding pro tento formát. Jako „výhodu“ můžeme uvést mnohonásobnou velikosti souborů s koncovkou BMP oproti JPEGu. U rastrové grafiky není problém narazit na obrázky, jejichž velikost dosahuje desítky Mb. Tím získáváme větší steganografickou kapacitu oproti ztrátovým formátům jako je JPEG. Mnoho steganografických systémů využívajících pro své ukrytí formát BMP, nepatří co do použitých metod mezi nejrobustnější a nejs sofistikovnější. I přesto jeho zařazení do diplomové práce je odůvodněno rozšířením v podvědomí většiny uživatelů výpočetní techniky, ve velké podpoře obrazových prohlížečů a dostupnosti steganografických nástrojů podporující tento formát.

3.1 Metody analyzačních postupů

Po rozsáhlé analýze dané problematiky, kterou stegoanalytickou metodu mám zvolit a jakým způsobem analyzovat nashromážděná data, byla zvolena metoda know-cover, při níž je k dispozici jak originální obrázek tak stegogram. Jako první ukazatel byla zvolena velikost obou souborů respektive jejich porovnání. Změna velikosti z logiky věci poukazuje na změnu obsahu daného souboru.

Do výsledné velikosti stegogramu se projevil jak použitý steganografický program tak formát obrázku. Zatímco u grafického formátu JPEG použitá aplikace Steghide nedopustila změnu v kladném či záporném směru nad cca 2,5 % velikosti coveru, aplikace Invisible secret byla mnohem benevolentnější. Ukryla do obrázku jpeg_10 o velikosti 462862 bytů nahrávku ve formátu mp3 o velikosti 245244 bytů. Výsledná změna činila „obrovských“

51 % což představuje výslednou velikost 701085 bytů. V průměru se ale u JPEG formátu změna velikosti pohybuje do 6 %.

Formát BMP byl v tomto ohledu zcela odlišný. Ve všech případech, kdy krycí obrázek pojal do svých útrob tajná data, nedošlo k nárůstu ani o jediný bit. To je způsobeno tím, že na rozdíl od formátu JPEG BMP neprochází žádným kódováním. Jak již bylo uvedeno, jedná se o bezeztrátový formát, jehož každý byte reprezentuje jeden pixel.

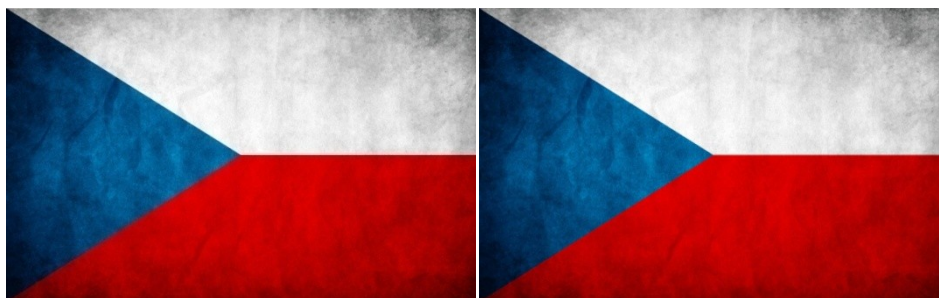
Druhou proměnnou, kterou jsem sledoval, byl počet barev v obrázku. Nejdříve jsem zapsal hodnoty počtu barev u originální nezměněných obrázků a následně po vložení jednotlivých ukryvaných dat.

U obrazového formátu JPEG nehrálo použití jednotlivých steganografických systémů významnou roli. Procentuální změny ze všech nasbíraných dat málokdy překročily hodnotu 1 % z počtu barev v originálním coveru. Ale i zde se setkávám s výjimkami. Steganografický systém Steghide měl u jednoho obrázku (jpeg_7), problém s ukrytím informací což jsem prvotně vyhodnotil jako zkreslení (převedení všech barev do odstínů fialové), které je způsobeno nízkým počtem barev v daném obrázku. U obrázků s mnohem nižším počtem barev k žádnému velkému zkreslení nedošlo. Obrázek 19 poukazuje na zmíněný fakt.



Obr. 19 – Zkreslený stegogram jpeg_7

Jpeg_7 vpravo je originální krycí obrázek s 166595 barvami. Obrázek vlevo jpeg_7_STEG_DOC je stegogram o 12274 barvách, který obsahuje ukrytá data ve formátu doc.

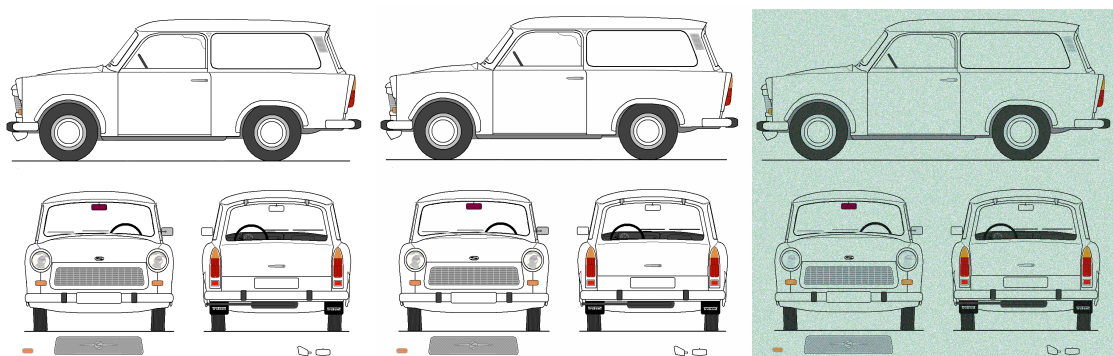


Obr. 20 – Nezkreslený stegogram jpeg_10 – originál dostupný z <http://good-wallpapers.com/places/10411>

Jpeg_10 vpravo je originální krycí obrázek s 21865 barvami. Obrázek vlevo jpeg_10_STEG_TXT je stegogram o 23893 barvách, který obsahuje ukrytá data ve formátu doc. Zde je sice zkreslení též patrné ale nikoliv v takové míře jako na předchozí fotografii.

K měření počtu barev jsem zmínil charakteristickou vlastnost program Invisible secret. Ten na rozdíl od všech ostatních aplikací během celého procesu ukrývání dat vůbec nezměnil počet barev stegogramu. To poukazuje, že se jedná o systém využívající Injection steganography method, kterou jsem popsal v kapitole 1.3.3.

Naprosto jiných výsledků bylo dosaženo při měření počtu barev u obrazového formátu BMP. Tři ze čtyř programů výrazným způsobem pozměnily tento počet a to v mnoha případech o dvou až třinásovek. Pohled do naměřených dat prozradí, že u tohoto formátu nezáleží tolik na použitém krycím coveru respektive na množství barev, ale velkou roli zde hraje použitý steganografický software. Steghide byl mezi ostatními aplikacemi nejúspěšnější (u formátu BMP), jeho procentuální zkreslení bylo pouze ve dvou případech větší než 4,5 %.



Obr. 21 – Ukázka stegogramů bmp_9 -
<http://www.trabime.cz/download/Trabant2.bmp>

V obrázku 21 jsou vloženy tři samostatné soubory. První zleva je originální obrázek (24 barev), druhý je stegogram (172 barev), který byl vytvořen v programu S-tools a obsahuje tajný soubor DOC (412672 bajtů). U tohoto obrázku je procentuální změna bitů 19,73 %. Třetí zleva je stegogram (88 barev), který obsahuje jako stegogram předchozí tajná data ve formátu DOC. Jeho zkresení je i přes menší rozdíl barev s originálem mnohem větší. Procentuální změna bitů je zde 46,79 %. Velikost všech třech grafických souborů je shodná a to 1404078 bajtů. Zde je názorně vidět jakou měrou se podílí na výsledném stegogramu zvolený software.

Poslední z charakteristik, která byla sledována na jednotlivých obrázcích jsou množství změněných bitů. Tedy kolik bitů se změnilo při porovnání originálu s již pozměněným obrázkem (stegogramem). I zde jsou mezi oběma obrazovými formáty značné rozdíly. Komprimovaný formát JPEG vkládá utajovaná data před samotným Huffmanovým kódováním, což je hlavní důvod proč procentuální počet změněných bitů dosahuje hodnot okolo 50 %. Přesněji aplikace Steghide, Steganos a JPEG Hide and seek dosahují hodnot v rozmezí 49 % ÷ 51 %. Program Invisible secret se v naměřených hodnotách liší pouze nepatrně, avšak hodnoty nad 50 % nikdy během testování nedosáhl, spíše se pohyboval v rozmezí 37 % ÷ 50 %.

U formátu BMP jsou naměřená data ve velké většině případů shodná pro všechny použité programy. Množství změněných bitů se pohybuje v rozmezí od 0,1 % do 6 %. V několika málo případech, kdy se naměřené hodnoty zvýší (10 % ÷ 20 %) jsou příčiny zřejmé. Velikost ukryvaných zpráv je v poměru ke krycímu coveru již nepřiměřená.

3.2 Analyzační software

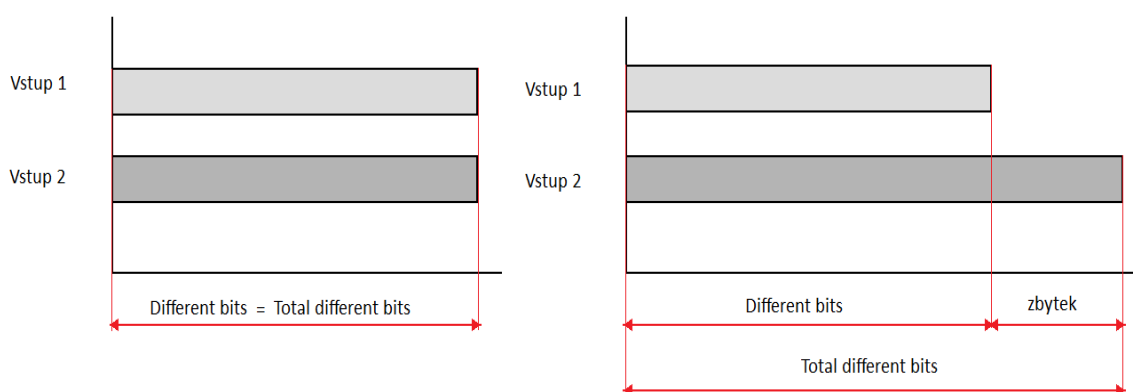
Prvně zmiňovanou měřenou charakteristiku, tedy velikost, lze pohodlně zjistit v operačním systému Windows označením daného souboru a kliknutím pravým tlačítkem myši vyvolat nabídku, v níž je položka vlastnosti. Zde je možné na záložce obecné zjistit velikost obrázku.

Druhou měřenou vlastností je počet barev. Pro tento účel byl zvolen bezplatně dostupný grafický prohlížeč a editor XnView. Jako mnoho dalších aplikací v této oblasti má možnost zobrazení histogramu nebo přímo vypíše počet barev. Podporuje více než 400 grafických formátů a nabízí mnoho nástrojů (změna velikosti, úprava světlosti a kontrastu, změna barevné hloubky, použití filtrů) pro správu a úpravu vašich obrázků a fotografií.

Třetí veličina, která poukazuje na možnost steganografické úpravy obrázků a která byla zaznamenána je změna bitů mezi originální a pozměněným obrázkem. Pro tento účel byl vybrán program s názvem Hexcmp. Ten kombinuje vlastnosti binárního komparátoru a hexadecimálního editoru. Data je možné dále modifikovat. Po delším seznámení bylo rozhodnuto, že aplikace nesplňuje požadavky a proto byl vytvořen program vlastní Bits Comparator.

3.2.1 Bits Comparator

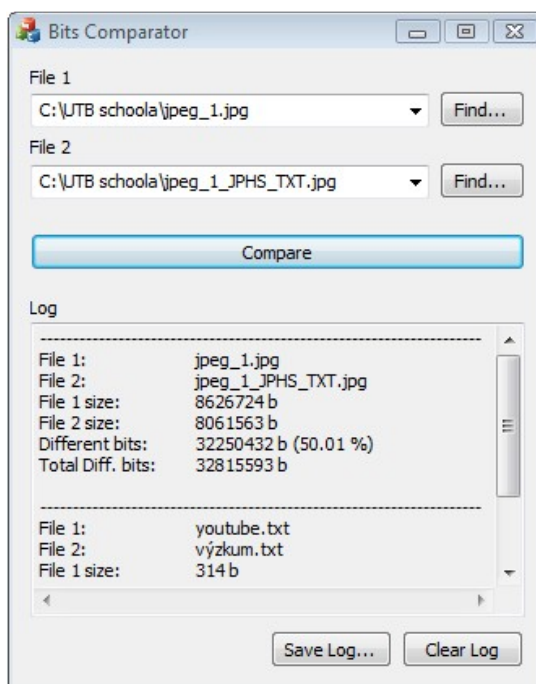
Bits Comparator je aplikace, která u dvou vstupních souborů vyhodnocuje počet změněných bitů. Program disponuje jednoduchým, avšak velice přehledným a intuitivním grafickým uživatelským rozhraním (obrázek číslo 23). Výstup je vypisován do okna, které je součástí rozhraní. Zde jsou vypsány užitečné informace, jako: názvy porovnávaných souborů, velikosti obou porovnávaných souborů a nakonec položky Different bits, což je označení počtu rozdílných bitů a Total different bits, který udává konečný počet rozdílných bitů. Tzn., že v případě kdy jsou porovnávány dva stejně velké soubory, které však mohou mít rozdílný obsah, nebude se u Different bits a Total different počet udávaný bitů lišit. V případě, že velikost vstupních souborů je rozdílná, budou tyto dva ukazatele udávat různá čísla. Different bits ukáže počet změněných bitů v pomyslně stejně velikých souborech. Total different bits ukáže Different bits + zbytek velikosti většího souboru. Tato skutečnost je graficky zpracována na obrázku 22.



Obr. 22 – Výstup programu Bits comparator

Velikou výhodou tohoto softwaru spatřuji v možnosti uložení výsledků komparace do textového souboru (.txt). Navíc není potřeba ukládat data po každém jednotlivém porovnání, ale v dialogovém okně se předchozí výsledky chronologicky řadí. K uložení

slouží tlačítko Save log., tlačítko Clear log., jak již název napovídá, celé dialogové okno smaže.



Obr. 23 - Uživatelské rozhraní programu Bits Comparator

Program je napsán v jazyce C++ s využitím knihovny MFC (Microsoft Foundation Classes). K vytvoření programu bylo použito vývojové prostředí Microsoft Visual Studio 2010. Data jsou načítána po blocích o velikosti 100 000 bajtů, které jsou následně porovnávány bajt po bajtu. Mezi porovnávanými bajty je nejdříve provedena operace XOR tzn., že výsledkem jsou na pozicích s rozdílnými hodnotami bitů logické „1“ a na pozicích se stejnými hodnotami logické „0“. Následně jsou v cyklu spočítány všechny logické „1“ v bajtu, a to tak, že se v každé iteraci provede bitový posun doprava, čímž se posune testovací bit na pozici prvního bitu a pomocí logického součinu s hodnotou 1 se vynulují všechny ostatní bity, takže výsledný byte může být buď nulový, nebo roven 1. Tato hodnota se přičítá k celkovému počtu rozdílných bitů.

Program byl vytvořen v rámci spolupráce se studentem UTB Bc. Richardem Červinkou, který se podílel na jeho vývoji. Zde je uvedena nejpodstatnější část zdrojového kódu. Celý je k nahlédnutí na DVD nosiči, který je přiložen k diplomové práci.

3.2.1.1 Ukázka zdrojového kódu

```

void CFileBitsCmprDlg::OnBnClickedBtnCompare()
{
    CWaitCursor waitCursor;

    CString file1Path;
    CString file2Path;
    m_comboFile1.GetWindowText(file1Path);
    m_comboFile2.GetWindowText(file2Path);
    CFile file1;
    CFile file2;

    //otevreni souboru
    if (file1Path == _T("") || file2Path == _T(""))
        return;

    if (!file1.Open(file1Path, CFile::modeRead | CFile::shareDenyWrite |
CFile::typeBinary | CFile::osSequentialScan))
    {
        AfxMessageBox(_T("Open file 1 error"));
        return;
    }

    if (!file2.Open(file2Path, CFile::modeRead | CFile::shareDenyWrite |
CFile::typeBinary | CFile::osSequentialScan))
    {
        AfxMessageBox(_T("Open file 2 error"));
        return;
    }

    //velikost souboru
    ULONGLONG file1Size = file1.GetLength();
    ULONGLONG file2Size = file2.GetLength();

    //zobrazeni progress baru, skryti tlacitka
    CRect btnRect;
    m_btnCompare.GetWindowRect(&btnRect);
    ScreenToClient(&btnRect);
    m_btnCompare.ShowWindow(SW_HIDE);
    m_progressBar.ShowWindow(SW_SHOW);
    m_progressBar.SetWindowPos(&wndTop, btnRect.left, btnRect.top,
btnRect.Width(), btnRect.Height(), SWP_NOZORDER);
    UpdateWindow();

    //porovnani velikosti souboru (spolecna cast)
    const ULONGLONG blockSize = 100000;
    ULONGLONG cmprSize = min(file1Size, file2Size);
    ULONGLONG cnt = cmprSize / blockSize;

    //parametry progressbaru
    m_progressBar.SetRange32(0, (int)cnt - 1);
    m_progressBar.SetPos(0);

    UINT totDif = 0;

```

```

BYTE f1Buff[blockSize];
BYTE f2Buff[blockSize];

//nacitani a porovnavani po blockSize
for (int i = 0; i < cnt; i++)
{
    file1.Read((void*)f1Buff, (UINT)blockSize);
    file2.Read((void*)f2Buff, (UINT)blockSize);
    totDif += Compare(f1Buff, f2Buff, blockSize);
    m_progressBar.SetPos(i);
}

//zbytek
UINT endSize = UINT(cmprSize - (cnt * blockSize));
file1.Read((void*)f1Buff, endSize);
file2.Read((void*)f2Buff, endSize);
totDif += Compare(f1Buff, f2Buff, endSize);
m_progressBar.SetPos(cnt - 1);

//vypis logu
CString tmpStr;
m_log.GetWindowText(tmpStr);
CString logStr;
logStr.Format(_T("-----\r\n"),
file1.GetFileName());
logStr.AppendFormat(_T("File 1: \t\t%s\r\n"), file1.GetFileName());
logStr.AppendFormat(_T("File 2: \t\t%s\r\n"), file2.GetFileName());
logStr.AppendFormat(_T("File 1 size: \t%d b\r\n"), file1Size);
logStr.AppendFormat(_T("File 2 size: \t%d b\r\n"), file2Size);
logStr.AppendFormat(_T("Different bits: \t%d b (%.2f %c)\r\n"), totDif,
((float)totDif / float(cmprSize * 8)) * 100.0f, _T('%'));
logStr.AppendFormat(_T("Total Diff. bits: \t%d b"), totDif + abs(long(file1Size -
file2Size)));
logStr.Append(_T("\r\n\r\n"));
logStr.Append(tmpStr);
m_log.SetWindowText(logStr);

file1.Close();
file2.Close();

//skryti progressbaru, zobrazeni tlacitka
m_progressBar.ShowWindow(SW_HIDE);
m_btnCompare.ShowWindow(SW_SHOW);
m_progressBar.SetPos(0);
}

UINT CFileBitsCmprDlg::Compare(BYTE *pB1, BYTE *pB2, UINT size)
{
    UINT totDif = 0;

    for (int i = 0; i < size; i++, pB1++, pB2++)

```

```

    {
        BYTE val = *pB1 ^ *pB2;
        for (int j = 0; j < 8; j++)
        {
            totDif += UINT((val >> j) & 1);        }}

        return totDif;        }
void CFileBitsCmprDlg::OnBnClickedButton1()
{
    CFileDialog dlg(TRUE, NULL, NULL, OFN_FILEMUSTEXIST, _T("All Files
(*.*)|*.*|"));
    if (dlg.DoModal() == IDOK)
    {
        m_comboFile1.InsertString(0, dlg.GetPathName());
        m_comboFile1.SetCurSel(0);

        if (m_comboFile1.GetCount() == 11)
            m_comboFile1.DeleteString(10);        }}

void CFileBitsCmprDlg::OnBnClickedButton2()
{
    CFileDialog dlg(TRUE, NULL, NULL, OFN_FILEMUSTEXIST, _T("All Files
(*.*)|*.*|"));
    if (dlg.DoModal() == IDOK)
    {
        m_comboFile2.InsertString(0, dlg.GetPathName());
        m_comboFile2.SetCurSel(0);

        if (m_comboFile2.GetCount() == 11)
            m_comboFile2.DeleteString(10);        }}

void CFileBitsCmprDlg::OnBnClickedButton3()
{
    m_log.SetWindowText(_T(""));        }

void CFileBitsCmprDlg::OnBnClickedBtnSaveLog()
{
    CFileDialog dlg(FALSE, _T("txt"), _T("*.txt"), OFN_HIDEREADONLY,
_T("TXT Files (*.txt)|*.txt|"));
    if (dlg.DoModal() == IDOK)
    {
        CString saveStr;
        m_log.GetWindowText(saveStr);
        CFile file;
        file.Open(dlg.GetPathName(), CFile::modeCreate | CFile::modeWrite |
CFile::shareExclusive);
        file.Write(saveStr.GetBuffer(), saveStr.GetLength() * sizeof(WCHAR));
        file.Close();
    }}

```

3.3 JPEG

Pro testování steganografického systému na grafický formát JPEG bylo použito 10 krycích obrázků. Jedná se o obrázky různých velikostí a různých obsahů. Jsou zde fotografie, které obsahují velké množství detailů (fraktální obrazec), či obrázky s malým počtem barev (vlajka). Jeden z krycích coverů je v černobílém vyhotovení s 8 bitovou hloubkou tzn., že je zde pouze 2^8 tedy 256 odstínů šedé barvy. Ukrývaná data byla ve formátu MP3, JPEG, DOC, PDF a TXT s rozdílnou velikostí viz. tabulka.

Všechny soubory jsou dostupné na DVD disku, který je přiložený k diplomové práci.

název obrázku	počet barev	celková velikost (bytů)	rozměry	bitů na pixel
jpeg_1	124988	8626724	4686x3713	24
jpeg_2	207797	4533632	4153x3492	24
jpeg_3	169710	3276764	3542x2660	24
jpeg_4	720552	2269509	4000x3000	24
jpeg_5	141770	1987120	2560x1600	24
jpeg_6	181141	1678017	1680x1050	24
jpeg_7	166595	1218853	2362x3550	24
jpeg_8	256	866616	2362x3550	8
jpeg_9	201098	635134	1600x1200	24
jpeg_10	21865	462862	1680x1050	24

Tab. 4 – Krycí obrázky ve formátu JPEG

name hiding data	formát	celková velikost (bytů)
MP3	.mp3	245244
JPEG	.jpg	191137
DOC	.doc	109056
PDF	.pdf	70381
TXT	.txt	33194

Tab. 5 – Ukrývaná data JPEG

Postupně byla vkládána jednotlivá ukrývaná data do krycích obrázků ($5 \times 10 = 50$). Tato operaci byla opakovaně provedena na již výše zmíněných programech a to konkrétně Steghide, Steganos, JPEG Hide and Seek a Invisible secret 4. Původně jsem zamýšlel použití i programu OpenPuff, ten ale vyhodnotil, že v žádném krycím obrázku ve formátu JPEG není dostatečná steganografická kapacita a proto jsem ho pro tuto část testování vynechal. U všech programů bylo použito shodné heslo a to: irbis.

Nejprve bylo použito programu Steghide, který nedisponuje jiným uživatelským rozhraním než příkazovou řádkou. Zprvu bylo zorientování a zapamatování jednotlivých příkazů komplikované. Nicméně nápověda, která se po spuštění programu zobrazí, byla naprosto

dostačující a schopnost příkazového řádku pamatovat si předchozí příkaz byla v tomto případě natolik vítána, že samotné steganografická operace spočívala v opětovném přepisování pouze několika symbolů. Velký nedostatek byl spatřen v častém ukončení embeddingu s chybovou hláškou „status_stack_overflow“ = přetečení zásobníku a „status_access_violation“ = porušení přístupu. Přesto, že se tyto chybové hlášky zobrazily hned v šestnácti případech, nejedná se o upozornění na nízkou steganografickou kapacitu. Tato zpráva byla prezentována jako „tho cover file is too short to embed the data“ (viz. obrázek 24). Program v některých případech dokázal zpracovávat data i po dobu 3 minut s nejasnou vidinou úspěchu. Nakonec se podařilo ukrýt data pouze do 11 obrázků.

```

C:\> Příkazový řádek

steghide version 0.5.1

the first argument must be one of the following:
embed, --embed          embed data
extract, --extract      extract data
info, --info            display information about a cover- or stego-file
info <filename>        display information about <filename>
encinfo, --encinfo      display a list of supported encryption algorithms
version, --version      display version information
license, --license      display steghide's license
help, --help            display this usage information

embedding options:
-ef, --embedfile        select file to be embedded
-ef <filename>          embed the file <filename>
-cf, --coverfile        select cover-file
-cf <filename>          embed into the file <filename>
-p, --passphrase        specify passphrase
-p <passphrase>         use <passphrase> to embed data
-sf, --stegofile        select stego file
-sf <filename>          write result to <filename> instead of cover-file
-e, --encryption        select encryption parameters
-e <a>[<m>]!<m>[<a>]    specify an encryption algorithm and/or mode
-e none                 do not encrypt data before embedding
-z, --compress          compress data before embedding (default)
-z <l>                  using level <l> (<1 best speed...9 best compression)
-Z, --dontcompress     do not compress data before embedding
-K, --nochecksum        do not embed crc32 checksum of embedded data
-N, --dontembedname    do not embed the name of the original file
-f, --force             overwrite existing files
-q, --quiet             suppress information messages
-v, --verbose           display detailed information

extracting options:
-sf, --stegofile        select stego file
-sf <filename>          extract data from <filename>
-p, --passphrase        specify passphrase
-p <passphrase>         use <passphrase> to extract data
-xf, --extractfile      select file name for extracted data
-xf <filename>          write the extracted data to <filename>
-f, --force             overwrite existing files
-q, --quiet             suppress information messages
-v, --verbose           display detailed information

options for the info command:
-p, --passphrase        specify passphrase
-p <passphrase>         use <passphrase> to get info about embedded data

To embed emb.txt in cvr.jpg: steghide embed -cf cvr.jpg -ef emb.txt
To extract embedded data from stg.jpg: steghide extract -sf stg.jpg

steghide embed -p irbis -cf jpeg_3.jpg -ef MP3.mp3 -sf jpeg_3_MP3.jpg
steghide: the cover file is too short to embed the data.

steghide embed -p irbis -cf jpeg_3.jpg -ef TXT.txt -sf jpeg_3_TXT.jpg
embedding "TXT.txt" in "jpeg_3.jpg"... done
   3 [main] steghide 3352 handle_exceptions: Exception: STATUS_ACCESS_VIOLATI
ON
   658 [main] steghide 3352 open_stackdumpfile: Dumping stack trace to steghide
.exe.stackdump

```

Obr. 24 – Steghide – hlášky

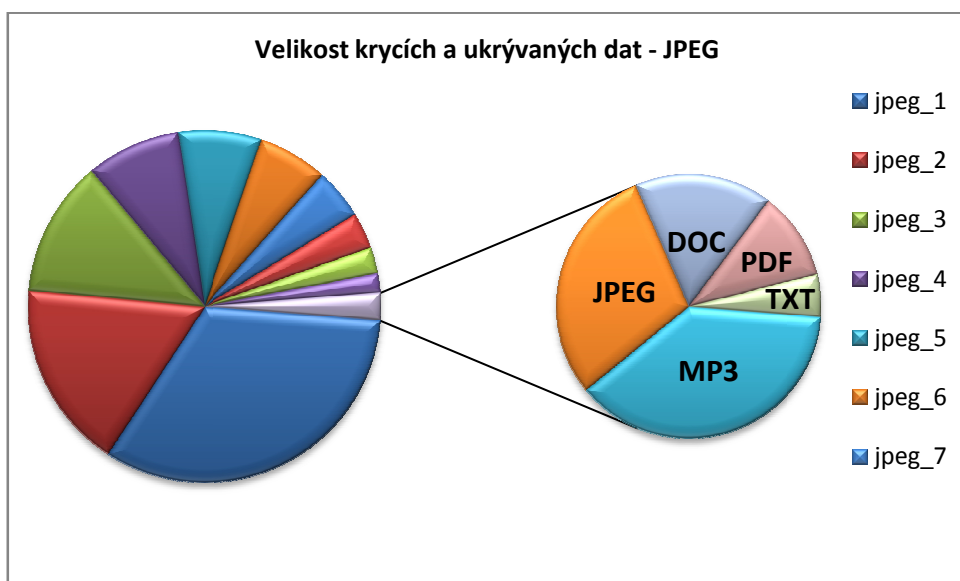
Program Steganos, který byl použit jako druhý, pracoval bezproblémově. Vše bylo zobrazováno v přehledných oknech. Pokud program vyhodnotil nedostatečnou steganografickou kapacitu uživatele okamžitě informoval a samotné vkládání se ukončilo. Aplikace Steganos pracoval velice rychle bez jediné chyby.

JPEG hide and seek byl další z řady steganografických nástrojů, který byl použit při testování. Bohužel aplikace měla několik nedostatků. Jako první a stěžejní problém považují nutnost restartu aplikace po každém embeddingu. V případě, že jsem tuto operaci neprovedl, celý program spadl. Pokoušel jsem spouštění i v režimu kompatibility, což nepřineslo tížený výsledek. Druhý nedostatek spatřuji v nezanesení podpory JPEG obrázku při osmibitové hloubce. V tomto případě jsem byl upozorněn na neznámý formát. Pro zabezpečení proti neoprávněnému uživateli jsem zvolil šifrovací algoritmus blowfish.

Poslední ze čtveřice softwarů byl Invisible secret 4. Z nabídky šifrovacích kódů bylo zvoleno AES Rijndael a již jednou zmíněné heslo: irbis. Software po celou dobu nevykazoval vůbec žádné chyby ani problémy, stejně jako druhý zpoplatněný program Steganos.

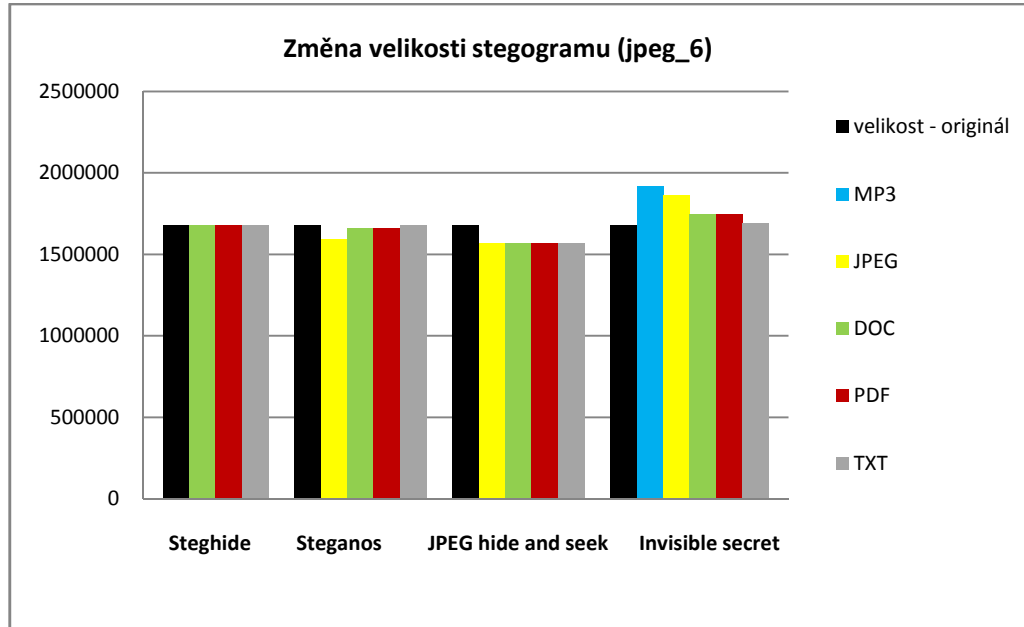
3.3.1 Zhodnocení JPEG

Veškerá nashromážděná data nalezneme v příloze diplomové práce a na DVD nosiči, který obsahuje excel soubory i s grafy. Pro představu, jakých hodnot bylo dosaženo, zde uvádím několik grafů.



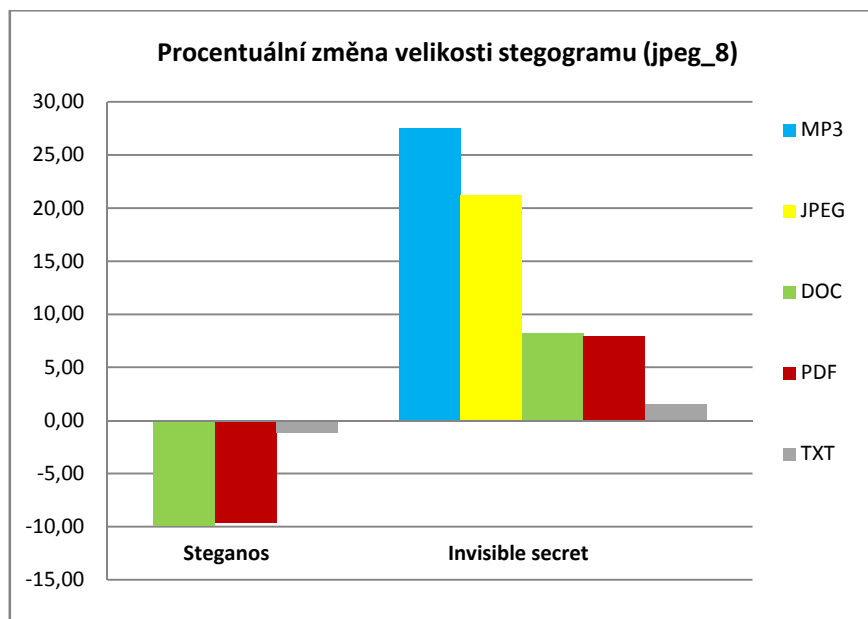
Obr. 25 - Velikost krycích a ukryvaných dat - JPEG

Tento graf (obr. 25) názorně zobrazuje data v tabulce 4 a 5. Tedy ukazuje poměr velikosti mezi krycími a ukrývanými daty.



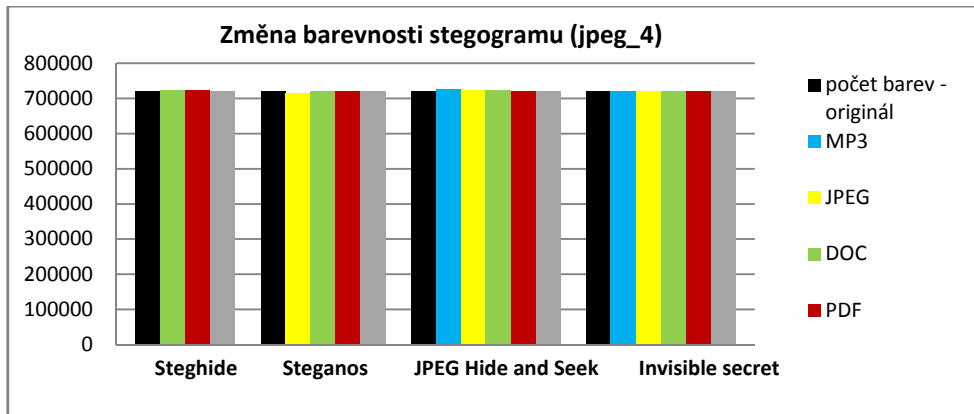
Obr. 26 - Změna velikosti stegogramu (jpeg_6)

Názorné zobrazení změny velikosti stegogramů vůči originálním krycím obrázkům (obr. 26).



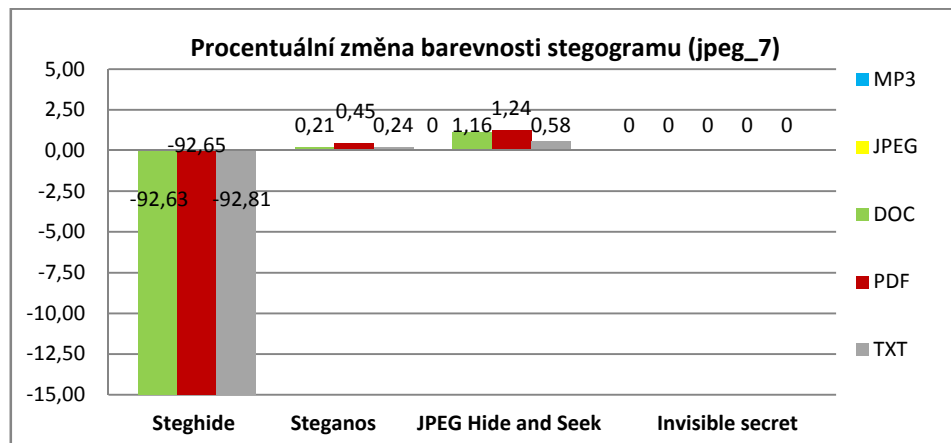
Obr. 27- Procentuální změna velikosti stegogramu (jpeg_8)

Graf (obr. 27) zobrazuje procentuální změnu velikosti stegogramu. Jak je zřejmé ze čtyř steganografických systémů si s obrázkem číslo 8 poradily pouze dva.



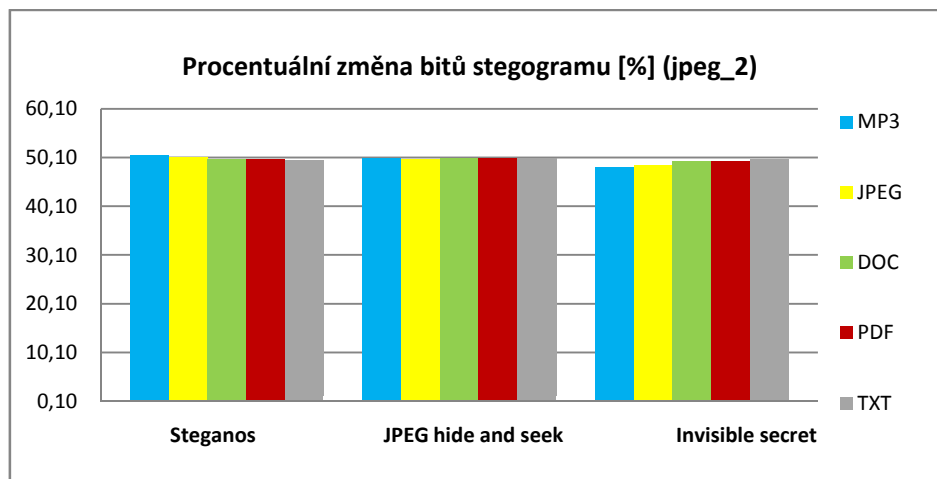
Obr. 28 - Změna barevnosti stegogramu (jpeg_4)

Na tomto grafu (obr. 28) je vidět, o kolik se změnil počet barev před a po ukrytí tajné informace.



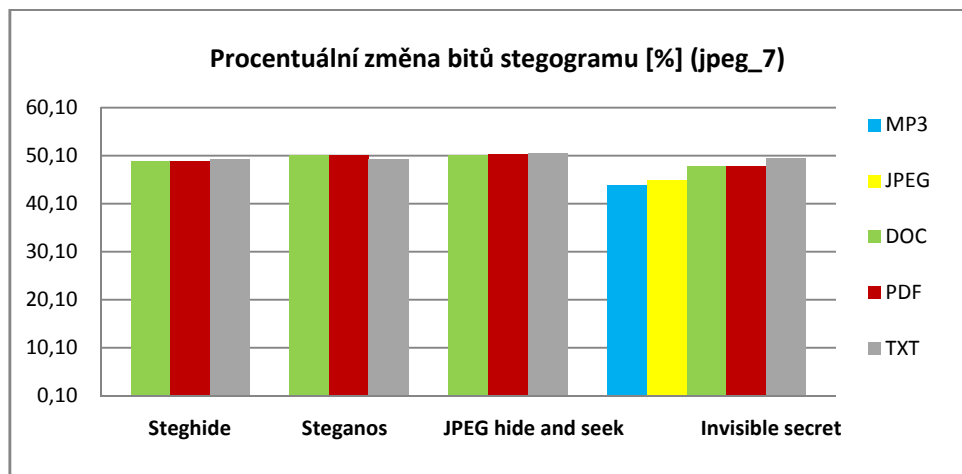
Obr. 29 - Procentuální změna barevnosti stegogramu (jpeg_7)

Zde je ukázka (obr. 29) shodné závislosti s předešlým grafem, pouze s tou změnou, že data jsou vyjádřena v procentech a hodnocen je jiný stegogram.



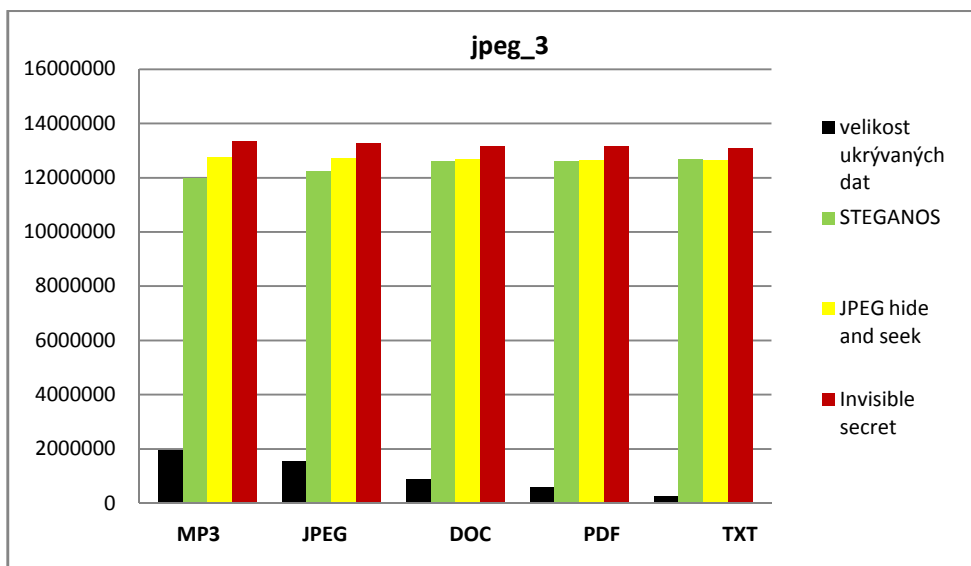
Obr. 30 - Procentuální změna bitů stegogramu (jpeg_2)

Zobrazení procentuálních změn bitů u stegogramu číslo 2. Aplikaci Steghide se do tohoto coveru nepodařilo uložit ani jednu tajnou informaci (obr. 30).



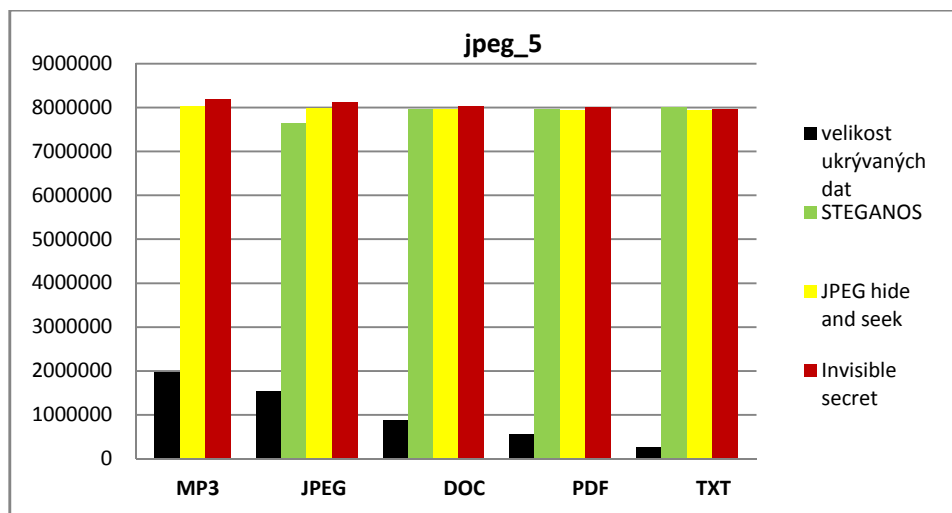
Obr. 31 - Procentuální změna bitů stegogramu (jpeg_7)

Graf (obr. 31) zobrazuje data jako graf předchozí pouze se změnou krycího media.



Obr. 32 – JPEG_3

Tento a následující graf (obr. 32-33) poukazuje na velikost ukrývaných dat v bitech a počet změněných bitů. Po odečtení z grafu vyplývá, že u tohoto konkrétního obrázku jsou ukrývaná data v sedmkrát až padesátkrát menší poměru než počet změněných bitů.



Obr. 33 – JPEG_5

Zde v tomto případě se poměr pohybuje od 4:1 do 30:1.

3.4 BMP

Stejně tak jako u formátu JPEG bylo i zde použito shodného počtu krycích obrázků i utajovaných informací. Podobný byl i výběr obsahu jednotlivých obrázků. Jako největší (z pohledu velikosti) z použitých krycích souborů byla fotografie dívky od profesionálního fotografa BcA. Daniela Vojtěcha. Tento totožný cover jsem použil dvakrát. V druhém případě byla změněna její bitová hloubka a to tak, že výsledkem je černobílý soubor s 256 barvami. Nechybí zde ani fraktální obrazec, fotografie se zaměřením na detail i panoráma zapadajícího slunce s množstvím detailů. Ukryvaná data byla taktéž podobná jako u JPEGu. Jedná se o DOC, JPEG, MP3, PDF a TXT. Podrobnosti lze vyčíst z následující tabulky nebo přímo z DVD nosiče, kde jsou všechny data uložena.

název obrázku	počet barev	celková velikost (bytů)	rozměry	bitů na pixel
bmp_1	166595	25162454	2362x3550	24
bmp_2	29388	12960054	2400x1800	24
bmp_3	256	8393278	2362x3550	8
bmp_4	42670	6912054	1920x1200	24
bmp_5	80750	5617950	2094x894	24
bmp_6	76641	3932214	1280x1024	24
bmp_7	37083	3000054	1000x1000	24
bmp_8	69960	1920054	100x640	24
bmp_9	24	1404078	1219x1150	8
bmp_10	30825	869814	604x480	24

Tab. 6 - Krycí obrázky ve formátu BMP

name hiding data	formát	celková velikost (bytů)
DOC	.doc	412672
JPEG	.jpg	231288
MP3	.mp3	152241
PDF	.pdf	89726
TXT	.txt	55776

Tab. 7 - *Ukrývaná data BMP*

Obdobně jako u předchozího formátu byla vkládána data do jednotlivých obrázků prostřednictvím uvedených aplikací. Heslo i v tomto případě zůstalo shodné a to: irbis. Aplikaci OpenPuff, která se neosvědčila u předchozího obrazového formátu, byla s úspěchem použita zde. Další použité programy byly Hide in Picture, S – Tools a již jedou použitý Steghide.

Prvně byla použita aplikaci Hide in Picture. Jednoduché rozhraní, snadné používání, a rychlost jsou základní rysy, které jsou pro tento program charakteristické. Jako šifrovací algoritmus byl zvolen blowfish s kontrolním zápisem CRC (cyclic redundancy check).

OpenPuff, jenž nebylo možnost otestovat u formátu JPEG, byl s větším či menším úspěchem vyzkoušel nyní. První komplikace se vyskytla v okamžiku, kdy běžně používané heslo bylo programem odmítnuto s odůvodněním minimální délky. Proto bylo heslo změněno na: snowleopard (pouze pro tento program). Abych mohl tento program použít v co největší míře, byla zaškrtnuta položka BIT SELECTION OPTIONS pro BMP formát na maximum. To zaručilo vměstnání největšího možného počtu bitů do jednotlivých obrázků. Poslední nedostatek spatřuji v odmítnutí hned dvou krycích coverů s odůvodněním nepodporovaného formátu. Jednalo se o bmp_3 (256 odstínů šedi) a bmp_9 (24 barev).

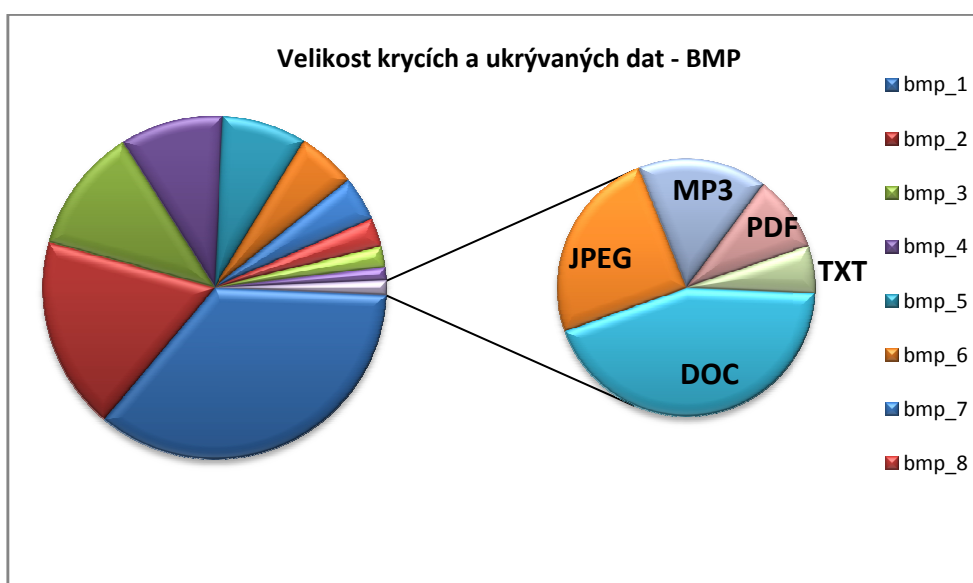
Předposlední aplikace pro formát BMP byl nástroj s názvem S – Tools. Ten mne hned od počátku zaujal svým netradičním ovládáním. Obrázky i ukrývaná data se do programu vkládaly přetažením a to se v praxi osvědčilo jako nesmírně promyšlené a velice rychlé ovládání. Program po celou dobu používání nevykazoval sebemenší chybu. Pouze čtyři neuložené stegogramy z padesáti možných považuji za výborný výsledek. Šifrování bylo nastaveno na algoritmus Idea.

Poslední, již jednou použitý software byl Steghide. I zde se objevovaly chybové hlášky ale pouze v pěti případech. V sedmnácti případech vyhodnotil program cover jako příliš

krátký, respektive ukryvaná data byla příliš objemná. Nicméně nad padesáti procentní úspěšnost je oproti 22 % u grafického formátu JPEG značné zlepšení.

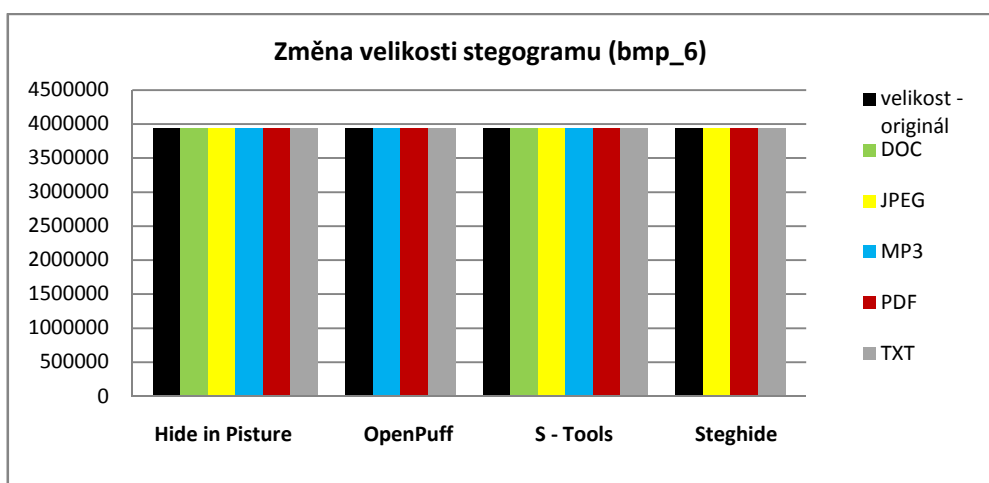
3.4.1 Zhodnocení BMP

Následující grafy slouží k zobrazení vybraných vztahů mezi programy, krycími médii a utajenými daty prostřednictvím naměřených hodnot. Kompletní seznam všech naměřených veličin je v příloze p II nebo na přiloženém DVD.



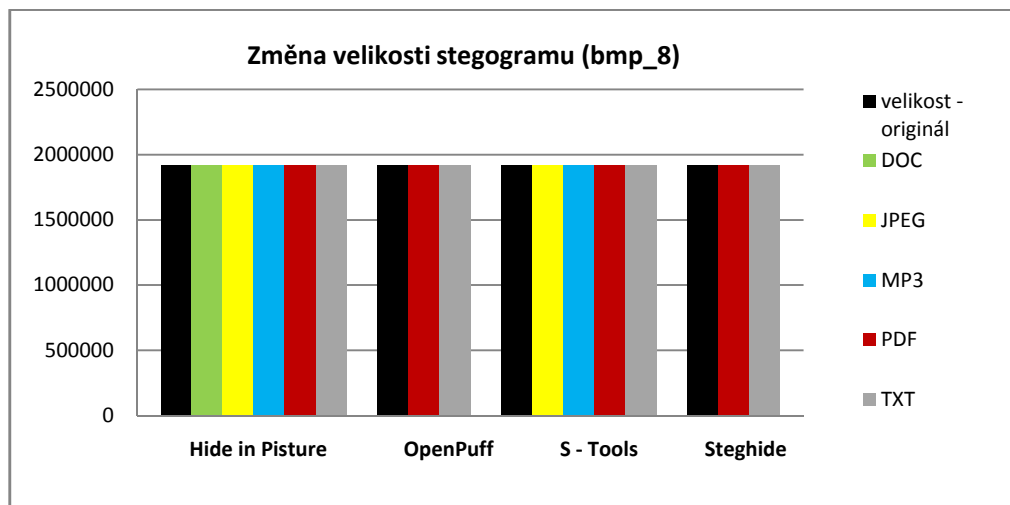
Obr. 34 - Velikost krycích a ukryvaných dat - BMP

Graf (obr. 34) znázorňující data z tabulky 6 a 7, tedy zobrazení poměru velikostí mezi krycími a ukryvanými daty.



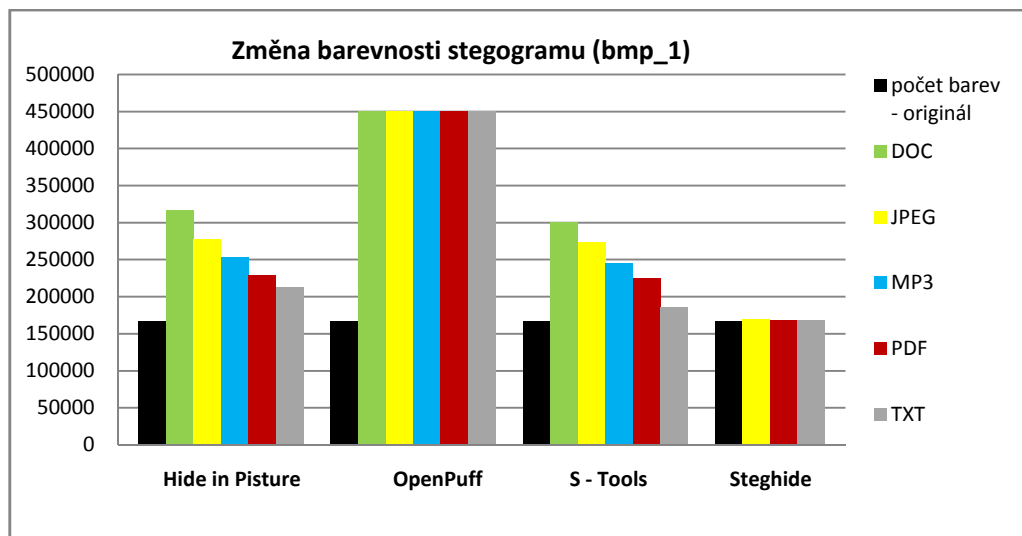
Obr. 35 - Změna velikosti stegogramu (bmp_6)

Na tomto grafu (obr. 35) je vidět fakt, o kterém již bylo psáno, a to že velikost stegogramu oproti originálu se u BMP formátu nemění.



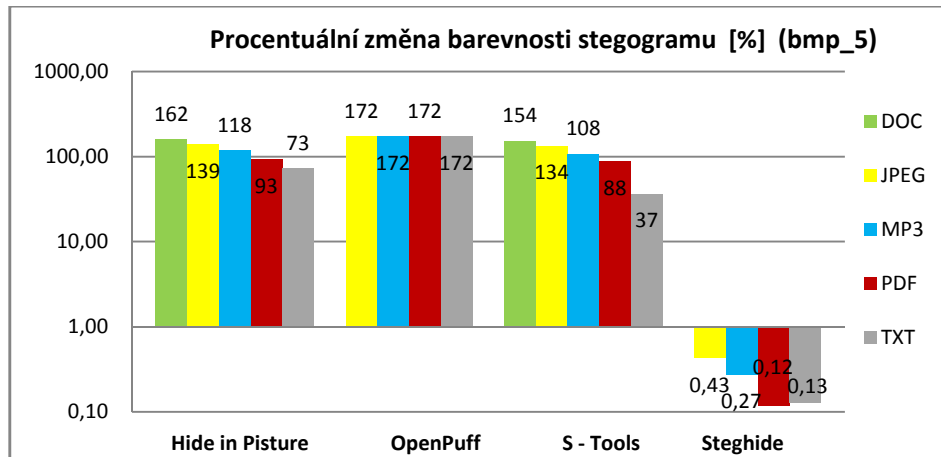
Obr. 36 - Změna velikosti stegogramu (bmp_8)

Tento graf (obr. 36) zobrazuje stejnou závislost jako graf předchozí pouze pro jiný obrázek.



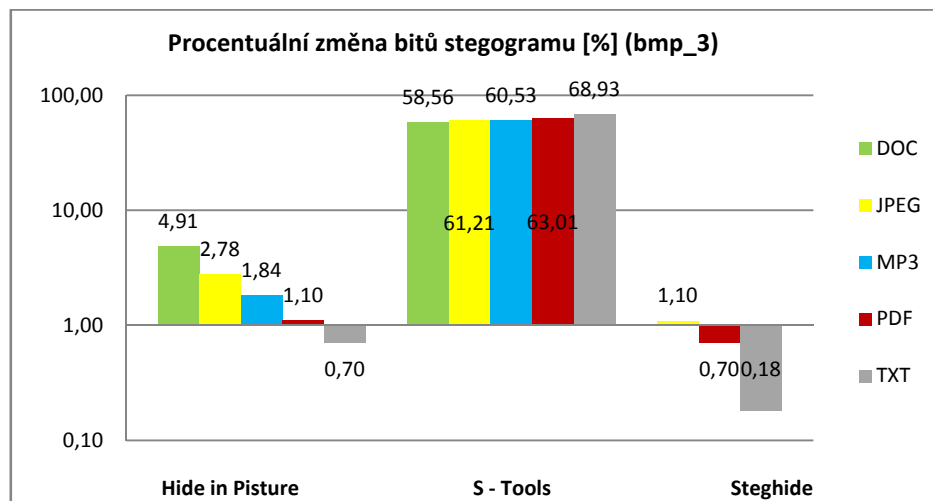
Obr. 37 - Změna barevnosti stegogramu (bmp_1)

Na grafu (obr. 37) je vidět jak se který program podepsal na změně počtu barev v coveru.



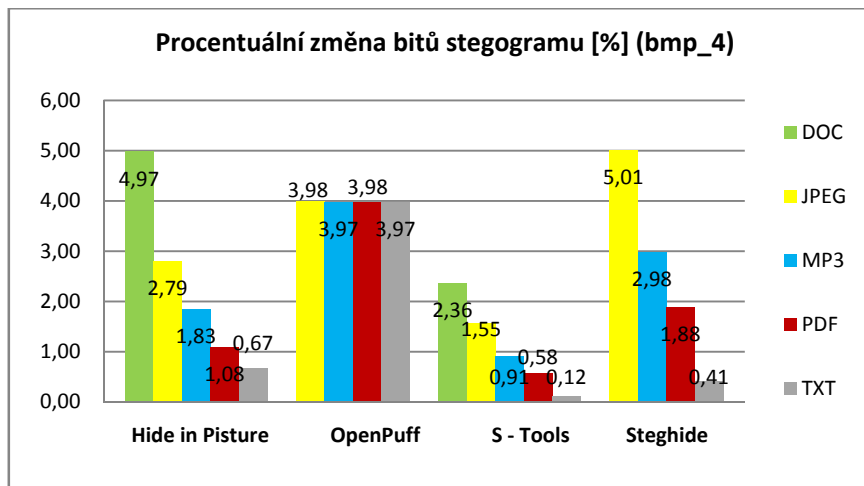
Obr. 38 - Procentuální změna barevnosti stegogramu (bmp_5)

Zde je vyobrazena procentuální změna barevnosti stegogramu. Graf (obr. 38) má horizontální osu vynesenu v logaritmském měřítku aby byla data lépe viditelná.



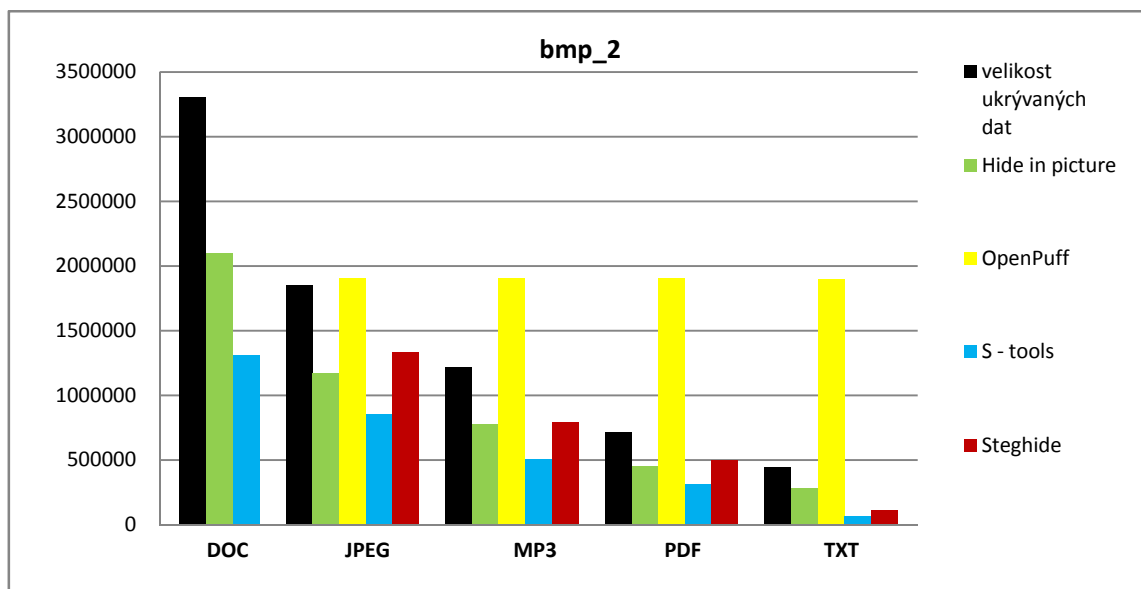
Obr. 39 - Procentuální změna bitů stegogramu (bmp_3)

Zobrazení procentuálních změn bitů u stegogramu číslo 3. Jak je zřejmé program OpenPuff si s tímto obrázkem neporadil (nepodporovaný formát). Opět zobrazeno v log. měřítku (obr 39).



Obr. 40 - Procentuální změna bitů stegogramu (bmp_4)

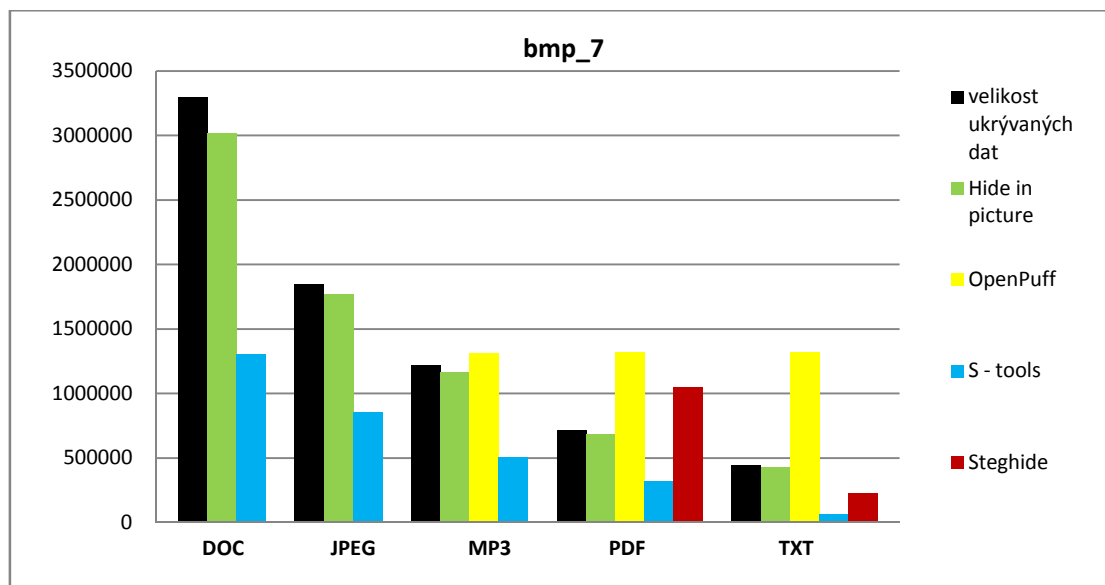
Graf (obr. 40) zobrazuje stejnou závislost jako graf předchozí pouze pro jiný obrázek.



Obr. 41 – BMP_2

Tento a následující graf (obr. 41-42) zobrazuje velikost ukryvaných dat v bitech a počet změněných bitů. Je zajímavé povšimnout si, že všechny aplikace vyjma OpenPuff, mají lineárně klesající průběh, zatímco zmiňovaný program se vyznačuje konstantním průběhem.

Jakou druhou podstatnou vlastnost, lze pozorovat při porovnání těchto posledních dvou grafů s posledními grafy obrazového formátu JPEG. Zatímco ukryvaná data do formátu BMP byla ve většině případů co se do velikosti dominantní (oproti počtu změněných bitů) v případě formátu JPEG je tomu přesně naopak.



Obr. 42 – BMP_7

3.5 Vyhodnocení programů

Neflexibilnější, nejspolehlivější a tedy nejvhodnější program, který byl v diplomové práci testován a jenž ukrývá data do obrazového formátu JPEG, je německý software Steganos. I když je jako jeden z mála zpoplatněný, jeho kvalita je očividná. Má přehledné grafické rozhraní a příjemně se používá. Navíc můžete očekávat jistou garanci profesionality, za kterou stojí německá organizace. Je možné, že stejně vhodný by byl i program JP Hide and Seek, ale jeho nestabilita ho řadí do pole poražených. Nesporně výborných hodnot dosahoval i program Steghide, který ale bohužel měl velké procento neúspěšných pokusů.

Jako nejvhodnější aplikace, která ukrývá data do BMP formátu tak, že běžný pozorovatel nemá šanci ničeho si povšimnout, není jedna. Rozhodl jsem se zvolit dva nejúspěšnější programy v této kategorii. Aplikace S – Tools je nejlepší z pohledu ovládání i spolehlivosti. Ale právě tyto dva důležité faktory chybí druhému z nich. Programu Steghide, který úspěšně ukryl data do 56 % všech obrazových medií. Tento výsledek není nejlepší a ovládání z příkazové řádky nejrychlejší, ale naměřené hodnoty dosahují jednoznačně nejlepších hodnot v tomto testu. Procentuální změna barevnosti či procentuální změna bitů je velice nízká většinou do 5 %. S – Tools zaostává hlavně v oblasti změny barevnosti, kde se hodnoty pohybují v oblasti dvojnásobku. Na druhou stranu oproti softwaru Steghide si dokázal lépe poradit s krycím coverem o 24 barvách jak si můžete přesvědčit na obrázku 21 nebo na DVD.

ZÁVĚR

V diplomové práci byla popsána historie steganografie i její současná digitální podoba, která skýtá nepředstavitelné možnosti. Dle mého názoru je tato kryptologická metoda neprávem opomíjena a její éra rozmachu ji teprve čeká. Vývoj sofistikovanějších steganografických metod nebo implementace do nejrůznějších krycích medií bude stegoanalytikům neustále ztěžovat jejich náročnou práci. Objem dat na Internetu stále roste a jen stěží jsou či budou všechna podrobena důkladné analýze. Této skutečnosti si bohužel povšiml, dnes již mrtvý zakladatel teroristické organizace al-Khaida, Usáma bin Muhammad bin Ládin. Ten pomocí steganografie organizoval hrůzný čin na Světové obchodní centrum v New Yorku 11. září 2001.

Teoretická část mé práce seznámila čtenáře s širokou problematikou steganografie. Podrobně byl popsán steganografický proces a názorně byly uvedeny nejběžnější metody ukrytí dat, které jsou založeny na principu změny bitů na pozici LSB – nejméně významného bitu. Další důležitá kapitola teoretické části práce byla věnována nejpoužívanějším krycím mediím a i zde je vidět jejich reálné použití. V závěru bylo popsáno, čím se zabývá stegoanalýza a jakých metod se při odhalování používá.

V praktické části bylo navázáno na část předcházející a to popisem vybraných sedmi steganografických aplikací, které byly následně podrobeny testování. Výstupem byla množina více jak tří set grafických souborů. Zvolený druh formátu byl reprezentován následujícími zástupci. JPEG zastupuje nejrozšířenějším typ obrazového formátu, zatímco nekomprimovaný formát s vysokou podporou z pohledu steganografie a grafických prohlížečů byl zvolen BMP. Nashromážděná data byla vyhodnocena podle parametru velikosti, dle počtu barev a podle počtu změněných bitů. Pro posledně jmenovanou charakteristiku bylo zapotřebí navrhnout a naprogramovat aplikaci, která počet změněných bitů vyhodnotí. Výsledná data byla prezentována formou přehledných grafů (v příloze i formou písemnou), kde jsou vidět základní souvislosti pro jednotlivá krycí media, ukryvaná data a v poslední řadě použité steganografické programy.

ZÁVĚR V ANGLIČTINĚ

The thesis describes the history and Steganography its present digital form, which offers unimaginable opportunities. In my opinion this cryptology method is unjustly neglected, and its era of expansion is still ahead. Development of more sophisticated methods steganography or implementation of the various media covering the stego-analyst constantly impedes their hard work. The volume of data on the Internet continues to grow and hardly any are or will be subjected to careful analysis. This fact, unfortunately, noticed now dead founder of the terrorist organization al-Khaida, Osama bin Muhammad bin Laden. The use of steganography organized a horrific act on the World Trade Center in New York 11th September 2001.

The theoretical part of my theses readers acquainted with the problems of steganography. I have described in detail steganography process and clearly we have shown the most common methods of concealing data, which is based on the change in the position bit LSB - least significant bit. Another important theoretical chapter I devoted most widely used cover media, and here you can see their real use. Finally, I described what deals steganalysis and what methods are used in the detection.

In the practical part, I followed the previous section by indicating steganography selected seven applications, which I subsequently underwent testing. The output is set over three hundred graphic files. The selected type of format is represented by the following representatives. Represents the most common type of JPEG image format, while an uncompressed format with high support from the perspective of steganography and graphical browsers was elected BMP. The collected data were evaluated according to the size parameters, depending on the number of colors and the number of bits changed. For the latter characteristic was necessary to design and program the application to evaluate the number of changed bits. The resulting data were presented in the form of graphs (in the Annex and in written form), where you can see the basic context for the various media covering, concealing data and ultimately used steganography programs.

SEZNAM POUŽITÉ LITERATURY

- [1] VONDRUŠKA, P. *Kryptologie, šifrování a tajná písma*. Praha: Albatros, 2006. ISBN 80-00-01888-8.
- [2] SINGH, S. *Kniha kódů a šifer „Utajování od starého Egypta po kvantovou kryptografii“*. Praha: Dokořán a Argo, 2009. ISBN 978-80-7363-268-7 (Dokořán), ISBN 978-80-257-0144-7.
- [3] PIPER, F. *Kryptografie*. Praha: Dokořán, 2006. ISBN: 80-7363-074-5.
- [4] ŽILKA, R. *Steganografie a stegoanalýza* [online]. 2008. 66 s. Diplomová práce. Masarykova univerzita. Vedoucí diplomové práce Ing. Mgr. Zdeněk Říha, Ph.D. [cit. 2010-03-01] Dostupné z WWW: <http://www.is.muni.cz/th/73058/fi_m/Steganografie_a_stegoanalýza.pdf>.
- [5] KADLEC, F. *Zpracování akustických signálů*. Praha: ČVUT, 2005. ISBN 80-01-02588-8.
- [6] FOLTÝNEK, T. *Komprimace a šifrování* [online]. 2010. 106 s. Mendelova univerzita v Brně. [cit. 2010-04-03] Dostupné z WWW: <https://akela.mendelu.cz/~foltynek/KAS/elearning/KAS_PDF.pdf>.
- [7] RYŠÁKOVÁ, A. *Steganografie I* [online]. 2003. Univerzita Hradec Králové. [cit. 2010-04-03] Dostupné z WWW: <<https://kryptologie.uhk.cz/81.htm>>.
- [8] MACIAK, Lukasz Grzegorz ; PONNIAH, Micheal Alexis; SHARMA, Renu. *MP3 STEGONOGRAPHY* [online]. 2005. [cit. 2010-05-16]. Dostupné z WWW: <<http://www.terminally-incoherent.com/stuff/projects/mp3stego/paper.doc>>.
- [9] ONDRUŠ, J. *Beztrátová komprese JPEG grafiky* [online]. 2009. 40 s. Diplomová práce. Univerzita Karlova v Praze. Vedoucí diplomové práce Mgr. Ján Lánský, [cit. 2010-03-01] Dostupné z WWW: <<http://paq8.hys.cz/jpeg.pdf>>.
- [10] ZIGULEC, Š. *Steganografická komunikace* [online]. 2006. 64 s. Diplomová práce. Univerzita Tomáše Bati ve Zlíně. Vedoucí diplomové práce Ing. Lubomír Macků, Ph.D. [cit. 2010-03-01] Dostupné z WWW: <http://http://dspace.knihovna.utb.cz/bitstream/handle/10563/2351/zigulec_2006_dp.pdf?sequence=1>.

- [11] THAMPI, SABU M. *Information Hiding Techniques: A Tutorial Review* [online]. 2004. 19 s. [cit. 2009-3-10] Dostupný z WWW: <<http://arxiv.org/ftp/arxiv/papers/0802/0802.3746.pdf>>.
- [12] BĚLONOHÝ, R., JEŽEK, F., a kol. *Steganografie 2* [online]. 2003. Univerzita Hradec Králové. [cit. 2010-04-03] Dostupné z WWW: <<https://kryptologie.uhk.cz/82.htm>>.
- [13] HOLOŠKA, J. *Odhalování steganografie pomocí neuronových sítí* [online]. 2008. 73 s. Diplomová práce. Univerzita Tomáše Bati ve Zlíně. Vedoucí diplomové práce Ing. Zuzana Oplatková, Ph.D. [cit. 2010-03-01] Dostupné z WWW: <http://dspace.knihovna.utb.cz/bitstream/handle/10563/5777/holo%C5%A1ka_2008_dp.pdf?sequence=1>.
- [14] JOHNOSON, N. F. *Information Hiding Steganography and Watermarking - Attacks and Countermeasures*. Hardcover: Springer, 2000. ISBN 978-0-7923-7204-2.
- [15] KATZENBEISSER, S. *Information hiding techniques for steganography and digital watermarking*. Hardcover: Artech House Books, 2000. ISBN 1-58053-035-4.
- [16] COLE, E. *Hiding in Plain Sight: Steganography and the Art of Covert Communication*. Wiley, 2003. ISBN: 0471444499.
- [17] PRATT, W. *Digital Image Processing*, Wiley, 2007, ISBN 978-0-471-76777-0.
- [18] WAYNER, P. *Disappearing Cryptography*, Morgan Kaufmann, 2002, ISBN 1558607692.
- [19] DAVERN, P., SCOTT, M. *Steganography: its history and its application to computer based data files*. [internal report] School of Computer Applications, Dublin City University, 1995. [cit. 2010-03-01] Dostupné z WWW: <<http://citeseer.ist.psu.edu/davern95steganography.html>>.
- [20] SONG, K. *Fast and Efficient Steganalysis Methods for Spread Spectrum Steganography*. [report] Department of Statistics, Florida State University, 2007. Dostupné z WWW: <<http://www.galaxy.gmu.edu/QMDNS2007/WebPages/PRES/CS3-1-Song.pdf>>.

SEZNAM POUŽITÝCH ZDROJŮ

Domovské a jiné stránky týkající se jednotlivých programů:

SPAMMIMIC – <http://www.spammimic.com/>

INVISIBLE SECRETS - <http://www.invisiblesecrets.com/>

STEGHIDE - <http://steghide.sourceforge.net/>

OPENPUFF -

http://members.fortunecity.it/blackvisionit/OpenPuff_Steganography_Home.html/

HIDE IN PICTURE - <http://hide-in-picture.sourceforge.net>

S – TOOLS - <http://members.cox.net/ebmmd/stego/stego/softwarewindows.html>

STEGANOS - <http://www.steganos.com/>

XNVIEW - <http://www.xnview.com/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ASCII	American Standard Code for Information Interchange
BBC	British Broadcasting Corporation
BMP	Windows Bitmap
CRC	Cyclic redundancy Cheb
CSPRNG	Cryptographically secure pseudo-random number generátor
DCT	Discrete cosine transform
GIF	Graphics Interchange Format
GNU (GPL)	GNU General Public License
HIP	Hide in picture
HTML	HyperText Markup Language
INV	Invisible secret 4
IT	Information technology
JPEG	Joint Photographic Experts Group
JPHS	JP hide and seek
LBE	Low bit encoding
LSB	Least significant bit
PUF	OpenPuff
RGB	Red, Green, Blue
RLE	Run length encoding
STEG	Steghide
TCP/IP	Transmission Control Protocol/Internet Protocol
TIFF	Tag Image File Format
TOOL	S – Tools
VLI	Variable Length Integer

SEZNAM OBRÁZKŮ

Obr. 1 – Rozdělení kryptografie.....	12
Obr. 2 – Cardanova mřížka	15
Obr. 3 – Struktura „information hiding“.....	18
Obr. 4 – Standardní paleta 256 odstínů šedé.....	22
Obr. 5 – Ovlivnění kvality obrázku ukládáním bitů na vyšší pozici v bajtu – dostupný z http://adela.utko.feec.vutbr.cz/mbis/prednaska/MBIS%2013.pdf	23
Obr. 6 – Hlavička protokolu TCP –dostupné z http://www.visi.com/~mjb/Drawings/	27
Obr. 7 – Hlavička rámce.....	28
Obr. 8 – James May, Road Test Yearbook Issue (1992)-dostupný z http://en.wikipedia.org/wiki/James_May	32
Obr. 9 - Maskovací audiogram. Maskující zvuk je šum o šíři pásma 90 Hz a středním kmitočtu 450 Hz s hladinami 20-80 dB, dle F. Kadlece ⁵	34
Obr. 10 - Časové oblasti současného maskování, před-maskování a dodatečného	34
Obr. 11 - Stegoanalýza dostupný: http://www.jjtc.com/Steganalysis/	38
Obr. 12 - – Uživatelské rozhraní programu Quick Stego	41
Obr. 13 – Uživatelské rozhraní programu JPHS.....	43
Obr. 14 – Uživatelské rozhraní programu OpenPuff.....	46
Obr. 15 - Uživatelské rozhraní programu Invisible secrets 4.....	48
Obr. 16 - Uživatelské rozhraní programu Steganos	49
Obr. 17 - Uživatelské rozhraní programu Hide In Picture 2.1.....	50
Obr. 18 - Uživatelské rozhraní programu S - Tools	51
Obr. 19 – Zkreslený stegogram jpeg_7.....	53
Obr. 20 – Nezkreslený stegogram jpeg_10 – originál dostupný z http://good-wallpapers.com/places/10411	54
Obr. 21 – Ukázka stegogramů bmp_9 - http://www.trabime.cz/download/Trabant2.bmp	54
Obr. 22 – Výstup programu Bits comparator	56
Obr. 23 - Uživatelské rozhraní programu Bits Comparator.....	57
Obr. 24 – Steghide – hlášky	62
Obr. 25 - Velikost krycích a ukryvaných dat - JPEG.....	63
Obr. 26 - Změna velikosti stegogramu (jpeg_6)	64
Obr. 27- Procentuální změna velikosti stegogramu (jpeg_8).....	64

Obr. 28 - Změna barevnosti stegogramu (jpeg_4).....	65
Obr. 29 - Procentuální změna barevnosti stegogramu (jpeg_7)	65
Obr. 30 - Procentuální změna bitů stegogramu (jpeg_2)	65
Obr. 31 - Procentuální změna bitů stenogramu (jpeg_7)	66
Obr. 32 – JPEG_3.....	66
Obr. 33 – JPEG_5.....	67
Obr. 34 - Velikost krycích a ukryvaných dat - BMP	69
Obr. 35 - Změna velikosti stegogramu (bmp_6)	69
Obr. 36 - Změna velikosti stegogramu (bmp_8)	70
Obr. 37 - Změna barevnosti stegogramu (bmp_1).....	70
Obr. 38 - Procentuální změna barevnosti stegogramu (bmp_5)	71
Obr. 39 - Procentuální změna bitů stegogramu (bmp_3)	71
Obr. 40 - Procentuální změna bitů stegogramu (bmp_4)	72
Obr. 41 – BMP_2	72
Obr. 42 – BMP_7	73

SEZNAM TABULEK

Tab. 1 – <i>Barvy a jejich číselné vyjádření</i>	22
Tab. 2 – <i>Význam bitů v hlavičce rámce formátu MP3</i>	29
Tab. 3 – <i>Metody steganografie</i>	37
Tab. 4 – <i>Krycí obrázky ve formátu JPEG</i>	61
Tab. 5 – <i>Ukrývaná data JPEG</i>	61
Tab. 6 - <i>Krycí obrázky ve formátu BMP</i>	67
Tab. 7 - <i>Ukrývaná data BMP</i>	68

SEZNAM PŘÍLOH

PŘÍLOHA P I: NAMĚŘENÉ HODNOTY JPEG

PŘÍLOHA P II: NAMĚŘENÉ HODNOTY BMP

PŘÍLOHA P I: NAMĚŘENÉ HODNOTY JPEG

název obrázku	počet barev	celková velikost (bytů)
jpeg_1	124988	8626724
jpeg_2	207797	4533632
jpeg_3	169710	3276764
jpeg_4	720552	2269509
jpeg_5	141770	1987120
jpeg_6	181141	1678017
jpeg_7	166595	1218853
jpeg_8	256	866616
jpeg_9	201098	635134
jpeg_10	21865	462862

name hiding data	formát	celková velikost (bytů)
MP3	.mp3	245244
JPEG	.jpg	191137
DOC	.doc	109056
PDF	.pdf	70381
TXT	.txt	33194

chyba I - status_stack_overflow

chyba II - status_access_violation

nelze - daný program vyhodnotil u krycího media nedostatečnou steganografickou kapacitu

nepodporovaný formát - software JPEG Hide & Seek neumí pracovat s 8 bitovým obrázkem

Formát zápisu dat: název obrázku; počet barev; procentuální změna barevnosti [%]; celková velikost (bytů); procentuální změna velikosti [%]; počet změněných bitů; procentuální změna bitů [%]

Steghide

jpeg_1_STEG_MP3; chyba II; jpeg_1_STEG_JPEG; chyba II; jpeg_1_STEG_DOC; chyba II; jpeg_1_STEG_PDF; chyba II; jpeg_1_STEG_TXT; chyba II; jpeg_2_STEG_MP3; chyba II; jpeg_2_STEG_JPEG; chyba II; jpeg_2_STEG_DOC; chyba II; jpeg_2_STEG_PDF; chyba II; jpeg_2_STEG_TXT; chyba II; jpeg_3_STEG_MP3; nelze; jpeg_3_STEG_JPEG; nelze; jpeg_3_STEG_DOC; chyba II; jpeg_3_STEG_PDF; chyba II; jpeg_3_STEG_TXT; chyba II; jpeg_4_STEG_MP3; nelze; jpeg_4_STEG_JPEG; nelze; jpeg_4_STEG_DOC; 723390; 0,39; 2258289; -0,49; 9040153; 50,04; jpeg_4_STEG_PDF; 723481; 0,41; 2257990; -0,51; 9035502; 50,02; jpeg_4_STEG_TXT; 721094; 0,08; 2254237; -0,67; 9028022; 50,06; jpeg_5_STEG_MP3;

nelze; jpeg_5_STEG_JPEG; nelze; jpeg_5_STEG_DOC; chyba II; jpeg_5_STEG_PDF; chyba II;
 jpeg_5_STEG_TXT; chyba II; jpeg_6_STEG_MP3; nelze; jpeg_6_STEG_JPEG; nelze;
 jpeg_6_STEG_DOC; 182016; 0,48; 1680783; 0,16; 6704436; 49,86; jpeg_6_STEG_PDF; 181955;
 0,45; 1680799; 0,17; 6706755; 49,88; jpeg_6_STEG_TXT; 181331; 0,10; 1678500; 0,03;
 6698631; 49,89; jpeg_7_STEG_MP3; nelze; jpeg_7_STEG_JPEG; nelze; jpeg_7_STEG_DOC;
 12274; -92,63; 1249595; 2,52; 4897162; 48,99; jpeg_7_STEG_PDF; 12239; -92,65; 1249258;
 2,49; 4895218; 48,98; jpeg_7_STEG_TXT; 11982; -92,81; 1237911; 1,56; 4885126; 49,33;
 jpeg_8_STEG_MP3; nelze; jpeg_8_STEG_JPEG; nelze; jpeg_8_STEG_DOC; nelze;
 jpeg_8_STEG_PDF; nelze; jpeg_8_STEG_TXT; nelze; jpeg_9_STEG_MP3; nelze;
 jpeg_9_STEG_JPEG; nelze; jpeg_9_STEG_DOC; nelze; jpeg_9_STEG_PDF; nelze;
 jpeg_9_STEG_TXT; 203416; 1,15; 628791; -1,00; 2516473; 50,03; jpeg_10_STEG_MP3; nelze;
 jpeg_10_STEG_JPEG; nelze; jpeg_10_STEG_DOC; nelze; jpeg_10_STEG_PDF; nelze;
 jpeg_10_STEG_TXT; 23893; 9,28; 464454; 0,34; 1850149; 49,79

Steganos

jpeg_1_STEGANOS_MP3;	124294;	-0,56;	8062880;	-6,54;	32691465;	50,68;
jpeg_1_STEGANOS_JPEG;	124540;	-0,36;	8127193;	-5,79;	32882887;	50,58;
jpeg_1_STEGANOS_DOC;	124862;	-0,10;	8215059;	-4,77;	33153720;	50,45;
jpeg_1_STEGANOS_PDF;	124851;	-0,11;	8217155;	-4,75;	33152219;	50,43;
jpeg_1_STEGANOS_TXT;	125002;	0,01;	8246739;	-4,40;	33245219;	50,39;
jpeg_2_STEGANOS_MP3;	197523;	-4,94;	4338520;	-4,30;	17531278;	50,51;
jpeg_2_STEGANOS_JPEG;	200703;	-3,41;	4422968;	-2,44;	17774578;	50,23;
jpeg_2_STEGANOS_DOC;	205795;	-0,96;	4541767;	0,18;	18097791;	49,81;
jpeg_2_STEGANOS_PDF;	205994;	-0,87;	4543276;	0,21;	18098168;	49,79;
jpeg_2_STEGANOS_TXT;	207458;	-0,16;	4583667;	1,10;	18140052;	49,47;
jpeg_3_STEGANOS_MP3;	168880;	-0,49;	2944429;	-10,14;	11974462;	50,84;
jpeg_3_STEGANOS_JPEG;	169848;	0,08;	3033080;	-7,44;	12239339;	50,44;
jpeg_3_STEGANOS_DOC;	170329;	0,36;	3147253;	-3,95;	12584433;	49,98;
jpeg_3_STEGANOS_PDF;	169961;	0,15;	3148199;	-3,92;	12588532;	49,98;
jpeg_3_STEGANOS_TXT;	169796;	0,05;	3184506;	-2,82;	12693754;	49,83;
jpeg_4_STEGANOS_MP3;	nelze;	jpeg_4_STEGANOS_JPEG;	716090;	-0,62;	2084616;	-8,15;
8487940;	50,90;	jpeg_4_STEGANOS_DOC;	719652;	-0,12;	2222823;	-2,06;
8906681;	50,09;	jpeg_4_STEGANOS_PDF;	720262;	-0,04;	2224255;	-1,99;
8913750;	50,09;	jpeg_4_STEGANOS_TXT;	720470;	-0,01;	2264029;	-0,24;
9033370;	49,87;	jpeg_5_STEGANOS_MP3;	nelze;	jpeg_5_STEGANOS_JPEG;	155846;	9,93;
1881043;	-5,34;	7638414;	50,76;	jpeg_5_STEGANOS_DOC;	148221;	4,55;
2004561;	0,88;	7971298;	49,71;			

jpeg_5_STEGANOS_PDF; 148132; 4,49; 2006171; 0,96; 7972666; 49,68;
jpeg_5_STEGANOS_TXT; 142827; 0,75; 2041543; 2,74; 8004654; 49,01;
jpeg_6_STEGANOS_MP3; nelze; jpeg_6_STEGANOS_JPEG; 179266; -1,04; 1591271; -5,17;
6439383; 50,58; jpeg_6_STEGANOS_DOC; 180812; -0,18; 1657840; -1,20; 6640281; 50,07;
jpeg_6_STEGANOS_PDF; 180748; -0,22; 1658398; -1,17; 6644353; 50,08;
jpeg_6_STEGANOS_TXT; 181091; -0,03; 1675809; -0,13; 6696656; 49,95;
jpeg_7_STEGANOS_MP3; nelze; jpeg_7_STEGANOS_JPEG; nelze; jpeg_7_STEGANOS_DOC;
166948; 0,21; 1130730; -7,23; 4534885; 50,13; jpeg_7_STEGANOS_PDF; 167345; 0,45;
1134114; -6,95; 4548450; 50,13; jpeg_7_STEGANOS_TXT; 166991; 0,24; 1206931; -0,98;
4769460; 49,40; jpeg_8_STEGANOS_MP3; nelze; jpeg_8_STEGANOS_JPEG; nelze;
jpeg_8_STEGANOS_DOC; 256; 0,00; 780291; -9,96; 3165560; 50,71; jpeg_8_STEGANOS_PDF;
256; 0,00; 783476; -9,59; 3174763; 50,65; jpeg_8_STEGANOS_TXT; 256; 0,00; 856788; -1,13;
3391385; 49,48; jpeg_9_STEGANOS_MP3; nelze; jpeg_9_STEGANOS_JPEG; nelze;
jpeg_9_STEGANOS_DOC; nelze; jpeg_9_STEGANOS_PDF; nelze; jpeg_9_STEGANOS_TXT;
200243; -0,43; 628168; -1,10; 2506750; 49,88; jpeg_10_STEGANOS_MP3; nelze;
jpeg_10_STEGANOS_JPEG; nelze; jpeg_10_STEGANOS_DOC; nelze;
jpeg_10_STEGANOS_PDF; nelze; jpeg_10_STEGANOS_TXT; 22748; 4,04; 453601; -2,00;
1820428; 50,17

JPEG Hide & Seek

jpeg_1_JPHS_MP3; 125282; 0,24; 8063088; -6,53; 32811719; 50,87; jpeg_1_JPHS_JPEG;
125214; 0,18; 8063426; -6,53; 32814668; 50,87; jpeg_1_JPHS_DOC; 125047; 0,05; 8062121; -
6,54; 32819601; 50,89; jpeg_1_JPHS_PDF; 125123; 0,11; 8062439; -6,54; 32817991; 50,88;
jpeg_1_JPHS_TXT; 125057; 0,06; 8061563; -6,55; 32815593; 50,88; jpeg_2_JPHS_MP3; 210411;
1,26; 4543173; 0,21; 18119796; 49,85; jpeg_2_JPHS_JPEG; 210159; 1,14; 4543993; 0,23;
18113319; 49,83; jpeg_2_JPHS_DOC; 208958; 0,56; 4537945; 0,10; 18103350; 49,87;
jpeg_2_JPHS_PDF; 209110; 0,63; 4538069; 0,10; 18100584; 49,86; jpeg_2_JPHS_TXT; 208413;
0,30; 4535378; 0,04; 18094616; 49,87; jpeg_3_JPHS_MP3; 171464; 1,03; 3159208; -3,59;
12755739; 50,47; jpeg_3_JPHS_JPEG; 171534; 1,07; 3153322; -3,77; 12734727; 50,48;
jpeg_3_JPHS_DOC; 169903; 0,11; 3128813; -4,52; 12664760; 50,60; jpeg_3_JPHS_PDF; 170140;
0,25; 3128854; -4,51; 12663829; 50,59; jpeg_3_JPHS_TXT; 169843; 0,08; 3121536; -4,74;
12638317; 50,61; jpeg_4_JPHS_MP3; 725882; 0,74; 2190096; -3,50; 8835358; 50,43;
jpeg_4_JPHS_JPEG; 722479; 0,27; 2192020; -3,41; 8842141; 50,42; jpeg_4_JPHS_DOC; 722555;
0,28; 2174878; -4,17; 8792360; 50,53; jpeg_4_JPHS_PDF; 721939; 0,19; 2174594; -4,18;
8793910; 50,55; jpeg_4_JPHS_TXT; 721260; 0,10; 2174355; -4,19; 8791292; 50,54;
jpeg_5_JPHS_MP3; 155818; 9,91; 2059791; 3,66; 8022157; 48,68; jpeg_5_JPHS_JPEG; 147851;

4,29; 2018488; 1,58; 7977905; 49,41; jpeg_5_JPHS_DOC; 142519; 0,53; 1992219; 0,26; 7956514; 49,92; jpeg_5_JPHS_PDF; 142407; 0,45; 1991588; 0,22; 7952533; 49,91; jpeg_5_JPHS_TXT; 142142; 0,26; 1987917; 0,04; 7945518; 49,96; jpeg_6_JPHS_MP3; nelze; jpeg_6_JPHS_JPEG; 182408; 0,70; 1568052; -6,55; 6379917; 50,86; jpeg_6_JPHS_DOC; 181606; 0,26; 1567415; -6,59; 6378991; 50,87; jpeg_6_JPHS_PDF; 181466; 0,18; 1567235; -6,60; 6377122; 50,86; jpeg_6_JPHS_TXT; 181299; 0,09; 1566761; -6,63; 6377661; 50,88; jpeg_7_JPHS_MP3; nelze; jpeg_7_JPHS_JPEG; nelze; jpeg_7_JPHS_DOC; 168532; 1,16; 1187260; -2,59; 4773413; 50,26; jpeg_7_JPHS_PDF; 168654; 1,24; 1180584; -3,14; 4754414; 50,34; jpeg_7_JPHS_TXT; 167557; 0,58; 1165577; -4,37; 4706993; 50,48; jpeg_8_JPHS_MP3; nepodporovaný formát; jpeg_8_JPHS_JPEG; nepodporovaný formát; jpeg_8_JPHS_DOC; nepodporovaný formát; jpeg_8_JPHS_PDF; nepodporovaný formát; jpeg_8_JPHS_TXT; nepodporovaný formát; jpeg_9_JPHS_MP3; nelze; jpeg_9_JPHS_JPEG; nelze; jpeg_9_JPHS_DOC; nelze; jpeg_9_JPHS_PDF; 207097; 2,98; 611683; -3,69; 2467573; 50,43; jpeg_9_JPHS_TXT; 203028; 0,96; 610933; -3,81; 2464256; 50,42; jpeg_10_JPHS_MP3; nelze; jpeg_10_JPHS_JPEG; nelze; jpeg_10_JPHS_DOC; nelze; jpeg_10_JPHS_PDF; nelze; jpeg_10_JPHS_TXT; 22053; 0,86; 460854; -0,43; 1843993; 50,02

Invisible secret 4

jpeg_1_INV_MP3; 124988; 0; 8864947; 2,76; 34741948; 48,99; jpeg_1_INV_JPEG; 124988; 0; 8810860; 2,13; 34685105; 49,21; jpeg_1_INV_DOC; 124988; 0; 8697712; 0,82; 34569507; 49,68; jpeg_1_INV_PDF; 124988; 0; 8695344; 0,80; 34569685; 49,70; jpeg_1_INV_TXT; 124988; 0; 8639946; 0,15; 34514369; 49,93; jpeg_2_INV_MP3; 207797; 0; 4771835; 5,25; 18342233; 48,05; jpeg_2_INV_JPEG; 207797; 0; 4717748; 4,06; 18279409; 48,43; jpeg_2_INV_DOC; 207797; 0; 4604600; 1,57; 18162807; 49,31; jpeg_2_INV_PDF; 207797; 0; 4602232; 1,51; 18162332; 49,33; jpeg_2_INV_TXT; 207797; 0; 4546834; 0,29; 18101600; 49,76; jpeg_3_INV_MP3; 169710; 0; 3514987; 7,27; 13338506; 47,43; jpeg_3_INV_JPEG; 169710; 0; 3460900; 5,62; 13283632; 47,98; jpeg_3_INV_DOC; 169710; 0; 3347752; 2,17; 13164859; 49,16; jpeg_3_INV_PDF; 169710; 0; 3345384; 2,09; 13168348; 49,20; jpeg_3_INV_TXT; 169710; 0; 3289986; 0,40; 13094775; 49,75; jpeg_4_INV_MP3; 720552; 0; 2507732; 10,50; 9248579; 46,10; jpeg_4_INV_JPEG; 720552; 0; 2453645; 8,11; 9195691; 46,85; jpeg_4_INV_DOC; 720552; 0; 2340497; 3,13; 9073305; 48,46; jpeg_4_INV_PDF; 720552; 0; 2338129; 3,02; 9077406; 48,53; jpeg_4_INV_TXT; 720552; 0; 2282731; 0,58; 9019227; 49,39; jpeg_5_INV_MP3; 141770; 0; 2225343; 11,99; 8187634; 45,99; jpeg_5_INV_JPEG; 141770; 0; 2171256; 9,27; 8129608; 46,80; jpeg_5_INV_DOC; 141770; 0; 2058108; 3,57; 8019549; 48,71; jpeg_5_INV_PDF; 141770; 0; 2055740; 3,45; 8014865; 48,73; jpeg_5_INV_TXT; 141770; 0; 2000342; 0,67; 7959661; 49,74; jpeg_6_INV_MP3; 181141; 0; 1916178; 14,19; 6943531; 45,30; jpeg_6_INV_JPEG; 181141; 0; 1862091; 10,97; 6889518; 46,25;

jpeg_6_INV_DOC; 181141; 0; 1748943; 4,23; 6776353; 48,43; jpeg_6_INV_PDF; 181141; 0; 1746575; 4,09; 6774645; 48,49; jpeg_6_INV_TXT; 181141; 0; 1691177; 0,78; 6716816; 49,65; jpeg_7_INV_MP3; 166595; 0; 1457076; 19,54; 5112588; 43,86; jpeg_7_INV_JPEG; 166595; 0; 1402989; 15,11; 5058692; 45,07; jpeg_7_INV_DOC; 166595; 0; 1289841; 5,82; 4943387; 47,91; jpeg_7_INV_PDF; 166595; 0; 1287473; 5,63; 4942956; 47,99; jpeg_7_INV_TXT; 166595; 0; 1232075; 1,08; 4885650; 49,57; jpeg_8_INV_MP3; 256; 0; 1104839; 27,49; 3702241; 41,89; jpeg_8_INV_JPEG; 256; 0; 1050752; 21,25; 3652130; 43,45; jpeg_8_INV_DOC; 256; 0; 937604; 8,19; 3536428; 47,15; jpeg_8_INV_PDF; 256; 0; 935236; 7,92; 3534476; 47,24; jpeg_8_INV_TXT; 256; 0; 879838; 1,53; 3474619; 49,36; jpeg_9_INV_MP3; 201098; 0; 873357; 37,51; 2744835; 39,29; jpeg_9_INV_JPEG; 201098; 0; 819270; 28,99; 2689315; 41,03; jpeg_9_INV_DOC; 201098; 0; 706122; 11,18; 2576846; 45,62; jpeg_9_INV_PDF; 201098; 0; 703754; 10,80; 2571721; 45,68; jpeg_9_INV_TXT; 201098; 0; 648356; 2,08; 2517200; 48,53; jpeg_10_INV_MP3; 21865; 0; 701085; 51,47; 2089837; 37,26; jpeg_10_INV_JPEG; 21865; 0; 646998; 39,78; 2035405; 39,32; jpeg_10_INV_DOC; 21865; 0; 533850; 15,34; 1921082; 44,98; jpeg_10_INV_PDF; 21865; 0; 531482; 14,83; 1919216; 45,14; jpeg_10_INV_TXT; 21865; 0; 476084; 2,86; 1862255; 48,90

PŘÍLOHA P II: NAMĚŘENÉ HODNOTY BMP

název obrázku	počet barev	celková velikost (bytů)
bmp_1	166595	25162454
bmp_2	29388	12960054
bmp_3	256	8393278
bmp_4	42670	6912054
bmp_5	80750	5617950
bmp_6	76641	3932214
bmp_7	37083	3000054
bmp_8	69960	1920054
bmp_9	24	1404078
bmp_10	30825	869814

name hiding data	formát	celková velikost (bytů)
DOC	.doc	412672
JPEG	.jpg	231288
MP3	.mp3	152241
PDF	.pdf	89726
TXT	.txt	55776

chyba I - status_stack_overflow

chyba II - status_access_violation

nelze - daný program vyhodnotil u krycího media nedostatečnou steganografickou kapacitu

nepodporovaný formát - software JPEG Hide & Seek neumí pracovat s 8 bitovým obrázkem

Formát zápisu dat: název obrázku; počet barev; procentuální změna barevnosti [%]; celková velikost (bytů); procentuální změna velikosti [%]; počet změněných bitů; procentuální změna bitů [%]

Hide In Picture

bmp_1_HIP_DOC; 316853; 90,19; 25162454; 0; 3157793; 1,57; bmp_1_HIP_JPEG; 276882; 66,20; 25162454; 0; 1775243; 0,88; bmp_1_HIP_MP3; 253414; 52,11; 25162454; 0; 1165675; 0,58; bmp_1_HIP_PDF; 228847; 37,37; 25162454; 0; 687760; 0,34; bmp_1_HIP_TXT; 212207; 27,38; 25162454; 0; 428341; 0,21; bmp_2_HIP_DOC; 44213; 50,45; 12960054; 0; 2095592; 2,02; bmp_2_HIP_JPEG; 41161; 40,06; 12960054; 0; 1174212; 1,13; bmp_2_HIP_MP3; 39065; 32,93; 12960054; 0; 774020; 0,75; bmp_2_HIP_PDF; 36768; 25,11; 12960054; 0; 455353; 0,44; bmp_2_HIP_TXT; 34963; 18,97; 12960054; 0; 282421; 0,27; bmp_3_HIP_DOC; 256; 0,00; 8393278; 0; 3299905; 4,91; bmp_3_HIP_JPEG; 256; 0,00; 8393278; 0; 1865647; 2,78;

bmp_3_HIP_MP3; 256; 0,00; 8393278; 0; 1235575; 1,84; bmp_3_HIP_PDF; 256; 0,00; 8393278; 0; 741659; 1,10; bmp_3_HIP_TXT; 256; 0,00; 8393278; 0; 471682; 0,70; bmp_4_HIP_DOC; 122188; 186,36; 6912054; 0; 2750381; 4,97; bmp_4_HIP_JPEG; 110187; 158,23; 6912054; 0; 1542195; 2,79; bmp_4_HIP_MP3; 100590; 135,74; 6912054; 0; 1014166; 1,83; bmp_4_HIP_PDF; 89202; 109,05; 6912054; 0; 598133; 1,08; bmp_4_HIP_TXT; 80203; 87,96; 6912054; 0; 372803; 0,67; bmp_5_HIP_DOC; 211919; 162,44; 5617950; 0; 3215677; 7,15; bmp_5_HIP_JPEG; 192647; 138,57; 5617950; 0; 1803854; 4,01; bmp_5_HIP_MP3; 175851; 117,77; 5617950; 0; 1189517; 2,65; bmp_5_HIP_PDF; 155735; 92,86; 5617950; 0; 700391; 1,56; bmp_5_HIP_TXT; 139808; 73,14; 5617950; 0; 435343; 0,97; bmp_6_HIP_DOC; 158413; 106,69; 3932214; 0; 3172721; 10,09; bmp_6_HIP_JPEG; 148488; 93,74; 3932214; 0; 1780352; 5,66; bmp_6_HIP_MP3; 137684; 79,65; 3932214; 0; 1171163; 3,72; bmp_6_HIP_PDF; 124238; 62,10; 3932214; 0; 691657; 2,20; bmp_6_HIP_TXT; 113222; 47,73; 3932214; 0; 429148; 1,36; bmp_7_HIP_DOC; 86056; 132,06; 3000054; 0; 3015735; 12,57; bmp_7_HIP_JPEG; 80332; 116,63; 3000054; 0; 1770169; 7,38; bmp_7_HIP_MP3; 75834; 104,50; 3000054; 0; 1164973; 4,85; bmp_7_HIP_PDF; 68896; 85,79; 3000054; 0; 685149; 2,85; bmp_7_HIP_TXT; 62613; 68,85; 3000054; 0; 427025; 1,78; bmp_8_HIP_DOC; 86661; 23,87; 1920054; 0; 2553816; 16,63; bmp_8_HIP_JPEG; 76102; 8,78; 1920054; 0; 1823185; 11,87; bmp_8_HIP_MP3; 75244; 7,55; 1920054; 0; 1198968; 7,81; bmp_8_HIP_PDF; 74268; 6,16; 1920054; 0; 708982; 4,62; bmp_8_HIP_TXT; 73141; 4,55; 1920054; 0; 438994; 2,86; bmp_9_HIP_DOC; 88; 266,67; 1404078; 0; 5255973; 46,79; bmp_9_HIP_JPEG; 58; 141,67; 1404078; 0; 4421071; 39,36; bmp_9_HIP_MP3; 40; 66,67; 1404078; 0; 3322046; 29,57; bmp_9_HIP_PDF; 39; 62,50; 1404078; 0; 1958784; 17,44; bmp_9_HIP_TXT; 40; 66,67; 1404078; 0; 1220416; 10,86; bmp_10_HIP_DOC; 121657; 294,67; 869814; 0; 2012498; 28,92; bmp_10_HIP_JPEG; 79253; 157,11; 869814; 0; 1337294; 19,22; bmp_10_HIP_MP3; 74105; 140,41; 869814; 0; 1030268; 14,81; bmp_10_HIP_PDF; 69267; 124,71; 869814; 0; 708886; 10,19; bmp_10_HIP_TXT; 65491; 112,46; 869814; 0; 440965; 6,34

OpenPuff

bmp_1_PUF_DOC; 450509; 170,42; 25162454; 0; 7954283; 3,95; bmp_1_PUF_JPEG; 450705; 170,54; 25162454; 0; 7953824; 3,95; bmp_1_PUF_MP3; 450523; 170,43; 25162454; 0; 7952537; 3,95; bmp_1_PUF_PDF; 450048; 170,14; 25162454; 0; 7956529; 3,95; bmp_1_PUF_TXT; 450492; 170,41; 25162454; 0; 7950467; 3,95; bmp_2_PUF_DOC; nelze; bmp_2_PUF_JPEG; 29311; -0,26; 12960054; 0; 1902704; 1,84; bmp_2_PUF_MP3; 29345; -0,15; 12960054; 0; 1902622; 1,84; bmp_2_PUF_PDF; 29286; -0,35; 12960054; 0; 1903117; 1,84; bmp_2_PUF_TXT; 29342; -0,16; 12960054; 0; 1900907; 1,83; bmp_3_PUF_DOC; nepodporovaný formát; bmp_3_PUF_JPEG; nepodporovaný formát; bmp_3_PUF_MP3; nepodporovaný formát;

bmp_3_PUF_PDF; nepodporovaný formát; bmp_3_PUF_TXT; nepodporovaný formát;
bmp_4_PUF_DOC; nelze; bmp_4_PUF_JPEG; 125375; 193,82; 6912054; 0; 2199550; 3,98;
bmp_4_PUF_MP3; 125287; 193,62; 6912054; 0; 2196575; 3,97; bmp_4_PUF_PDF; 125413;
193,91; 6912054; 0; 2198972; 3,98; bmp_4_PUF_TXT; 125344; 193,75; 6912054; 0; 2197334;
3,97; bmp_5_PUF_DOC; nelze; bmp_5_PUF_JPEG; 219930; 172,36; 5617950; 0; 2531642; 5,63;
bmp_5_PUF_MP3; 219863; 172,28; 5617950; 0; 2535111; 5,64; bmp_5_PUF_PDF; 219986;
172,43; 5617950; 0; 2532823; 5,64; bmp_5_PUF_TXT; 219858; 172,27; 5617950; 0; 2533879;
5,64; bmp_6_PUF_DOC; nelze; bmp_6_PUF_JPEG; nelze; bmp_6_PUF_MP3; 153671; 100,51;
3932214; 0; 1352734; 4,30; bmp_6_PUF_PDF; 153619; 100,44; 3932214; 0; 1355523; 4,31;
bmp_6_PUF_TXT; 153637; 100,46; 3932214; 0; 1355481; 4,31; bmp_7_PUF_DOC; nelze;
bmp_7_PUF_JPEG; nelze; bmp_7_PUF_MP3; 81994; 17,20; 3000054; 0; 1315620; 5,48;
bmp_7_PUF_PDF; 82064; 121,30; 3000054; 0; 1317443; 5,49; bmp_7_PUF_TXT; 82147; 121,52;
3000054; 0; 1317474; 5,49; bmp_8_PUF_DOC; nelze; bmp_8_PUF_JPEG; nelze;
bmp_8_PUF_MP3; nelze; bmp_8_PUF_PDF; 75980; 104,89; 1920054; 0; 864833; 5,63;
bmp_8_PUF_TXT; 75855; 104,55; 1920054; 0; 865008; 5,63; bmp_9_PUF_DOC; nelze;
bmp_9_PUF_JPEG; nelze; bmp_9_PUF_MP3; nelze; bmp_9_PUF_PDF; nelze;
bmp_9_PUF_TXT; nelze; bmp_10_PUF_DOC; nelze; bmp_10_PUF_JPEG; nelze;
bmp_10_PUF_MP3; nelze; bmp_10_PUF_PDF; nelze; bmp_10_PUF_TXT; nelze

S - Tools

bmp_1_TOOL_DOC; 300213; 80,21; 25162454; 0; 1306082; 0,65; bmp_1_TOOL_JPEG; 272986;
63,86; 25162454; 0; 858118; 0,43; bmp_1_TOOL_MP3; 244798; 46,94; 25162454; 0; 504251;
0,25; bmp_1_TOOL_PDF; 225008; 35,06; 25162454; 0; 317236; 0,16; bmp_1_TOOL_TXT;
185641; 11,43; 25162454; 0; 64739; 0,03; bmp_2_TOOL_DOC; 42901; 45,98; 12960054; 0;
1307601; 1,26; bmp_2_TOOL_JPEG; 40547; 37,97; 12960054; 0; 857149; 0,83;
bmp_2_TOOL_MP3; 38116; 29,70; 12960054; 0; 504521; 0,49; bmp_2_TOOL_PDF; 36184;
23,13; 12960054; 0; 317167; 0,31; bmp_2_TOOL_TXT; 31772; 8,11; 12960054; 0; 65058; 0,06;
bmp_3_TOOL_DOC; 256; 0,00; 8393278; 0; 39320157; 58,56; bmp_3_TOOL_JPEG; 256; 0,00;
8393278; 0; 41102410; 61,21; bmp_3_TOOL_MP3; 251; -1,95; 8393278; 0; 40644403; 60,53;
bmp_3_TOOL_PDF; 254; -0,78; 8393278; 0; 42308896; 63,01; bmp_3_TOOL_TXT; 256; 0,00;
8393278; 0; 46281987; 68,93; bmp_4_TOOL_DOC; 115840; 171,48; 6912054; 0; 1307614; 2,36;
bmp_4_TOOL_JPEG; 106458; 149,49; 6912054; 0; 857949; 1,55; bmp_4_TOOL_MP3; 95072;
122,81; 6912054; 0; 503765; 0,91; bmp_4_TOOL_PDF; 86213; 102,05; 6912054; 0; 318403; 0,58;
bmp_4_TOOL_TXT; 61685; 44,56; 6912054; 0; 64714; 0,12; bmp_5_TOOL_DOC; 205175;
154,09; 5617950; 0; 1306903; 2,91; bmp_5_TOOL_JPEG; 188998; 134,05; 5617950; 0; 857971;
1,91; bmp_5_TOOL_MP3; 168238; 108,34; 5617950; 0; 502967; 1,12; bmp_5_TOOL_PDF;

151594; 87,73; 5617950; 0; 317817; 0,71; bmp_5_TOOL_TXT; 110308; 36,60; 5617950; 0; 64816; 0,14; bmp_6_TOOL_DOC; 156908; 104,73; 3932214; 0; 1306571; 4,15; bmp_6_TOOL_JPEG; 147134; 91,98; 3932214; 0; 857732; 2,73; bmp_6_TOOL_MP3; 133211; 73,81; 3932214; 0; 503852; 1,60; bmp_6_TOOL_PDF; 121460; 58,48; 3932214; 0; 318186; 1,01; bmp_6_TOOL_TXT; 93473; 21,96; 3932214; 0; 64709; 0,21; bmp_7_TOOL_DOC; 83179; 124,30; 3000054; 0; 1307617; 5,45; bmp_7_TOOL_JPEG; 79834; 115,28; 3000054; 0; 857883; 3,57; bmp_7_TOOL_MP3; 73269; 97,58; 3000054; 0; 503462; 2,10; bmp_7_TOOL_PDF; 67244; 81,33; 3000054; 0; 318043; 1,33; bmp_7_TOOL_TXT; 50386; 35,87; 3000054; 0; 65074; 0,27; bmp_8_TOOL_DOC; nelze; bmp_8_TOOL_JPEG; 75993; 8,62; 1920054; 0; 858244; 5,59; bmp_8_TOOL_MP3; 75026; 7,24; 1920054; 0; 503510; 3,28; bmp_8_TOOL_PDF; 73872; 5,59; 1920054; 0; 317287; 2,07; bmp_8_TOOL_TXT; 71216; 1,80; 1920054; 0; 64807; 0,42; bmp_9_TOOL_DOC; 172; 616,67; 1404078; 0; 2216518; 19,73; bmp_9_TOOL_JPEG; 164; 583,33; 1404078; 0; 1876248; 16,70; bmp_9_TOOL_MP3; 157; 554,17; 1404078; 0; 1578365; 14,05; bmp_9_TOOL_PDF; 142; 491,67; 1404078; 0; 1489399; 13,26; bmp_9_TOOL_TXT; 109; 354,17; 1404078; 0; 1167454; 10,39; bmp_10_TOOL_DOC; nelze; bmp_10_TOOL_JPEG; nelze; bmp_10_TOOL_MP3; nelze; bmp_10_TOOL_PDF; 68576; 122,47; 869814; 0; 317597; 4,56; bmp_10_TOOL_TXT; 50629; 64,25; 869814; 0; 64961; 0,93

Steghide

bmp_1_STEG_DOC; chyba I; bmp_1_STEG_JPEG; 169421; 1,70; 25162454; 0; 2654293; 1,32; bmp_1_STEG_MP3; chyba II; bmp_1_STEG_PDF; 168467; 1,12; 25162454; 0; 1002477; 0,50; bmp_1_STEG_TXT; 167589; 0,60; 25162454; 0; 218595; 0,11; bmp_2_STEG_DOC; chyba II; bmp_2_STEG_JPEG; 30430; 3,55; 12960054; 0; 1333932; 1,29; bmp_2_STEG_MP3; 30221; 2,83; 12960054; 0; 795039; 0,77; bmp_2_STEG_PDF; 30032; 2,19; 12960054; 0; 499902; 0,48; bmp_2_STEG_TXT; 29715; 1,11; 12960054; 0; 108237; 0,10; bmp_3_STEG_DOC; chyba II; bmp_3_STEG_JPEG; chyba II; bmp_3_STEG_MP3; 256; 0,00; 8393278; 0; 735926; 1,10; bmp_3_STEG_PDF; 256; 0,00; 8393278; 0; 472631; 0,70; bmp_3_STEG_TXT; 256; 0,00; 8393278; 0; 120472; 0,18; bmp_4_STEG_DOC; nelze; bmp_4_STEG_JPEG; 44183; 3,55; 6912054; 0; 2768267; 5,01; bmp_4_STEG_MP3; 43776; 2,59; 6912054; 0; 1647129; 2,98; bmp_4_STEG_PDF; 43526; 2,01; 6912054; 0; 1039946; 1,88; bmp_4_STEG_TXT; 43048; 0,89; 6912054; 0; 229441; 0,41; bmp_5_STEG_DOC; nelze; bmp_5_STEG_JPEG; 81095; 0,43; 5617950; 0; 2532997; 5,64; bmp_5_STEG_MP3; 80970; 0,27; 5617950; 0; 1515671; 3,37; bmp_5_STEG_PDF; 80844; 0,12; 5617950; 0; 965747; 2,15; bmp_5_STEG_TXT; 80853; 0,13; 5617950; 0; 219844; 0,49; bmp_6_STEG_DOC; nelze; bmp_6_STEG_JPEG; nelze; bmp_6_STEG_MP3; 79953; 4,32; 3932214; 0; 1635082; 5,20; bmp_6_STEG_PDF; 79409; 3,61; 3932214; 0; 1034018; 3,29; bmp_6_STEG_TXT; 78094; 1,90; 3932214; 0; 224117; 0,71;

bmp_7_STEG_DOC; nelze; bmp_7_STEG_JPEG; nelze; bmp_7_STEG_MP3; nelze;
bmp_7_STEG_PDF; 37557; 1,28; 3000054; 0; 1049720; 4,37; bmp_7_STEG_TXT; 37342; 0,70;
3000054; 0; 230907; 0,96; bmp_8_STEG_DOC; nelze; bmp_8_STEG_JPEG; nelze;
bmp_8_STEG_MP3; nelze; bmp_8_STEG_PDF; 70061; 0,14; 1920054; 0; 873606; 5,69;
bmp_8_STEG_TXT; 69975; 0,02; 1920054; 0; 202262; 1,32; bmp_9_STEG_DOC; nelze;
bmp_9_STEG_JPEG; nelze; bmp_9_STEG_MP3; nelze; bmp_9_STEG_PDF; 51; 112,50;
1404078; 0; 933771; 8,31; bmp_9_STEG_TXT; 45; 87,50; 1404078; 0; 194638; 1,73;
bmp_10_STEG_DOC; nelze; bmp_10_STEG_JPEG; nelze; bmp_10_STEG_MP3; nelze;
bmp_10_STEG_PDF; nelze; bmp_10_STEG_TXT; 31014; 0,61; 869814; 0; 220460; 3,17