SLOVENSKÁ TECHNICKÁ UNIVERZITA
V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

STEGANOGRAFIA V OBRAZE A VIDEU

DIZERTAČNÁ PRÁCA

2011 Matúš Jókay

SLOVENSKÁ TECHNICKÁ UNIVERZITA
V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

STEGANOGRAFIA V OBRAZE A VIDEU

DIZERTAČNÁ PRÁCA

Evidenčné číslo:	FEI-12306-10139
Študijný program:	Aplikovaná informatika
Študijný odbor (kód a názov):	9.2.9 Aplikovaná informatika
Pracovisko:	Ústav informatiky a matematiky
Školiteľ:	Prof. RNDr. Otokar Grošek, PhD.

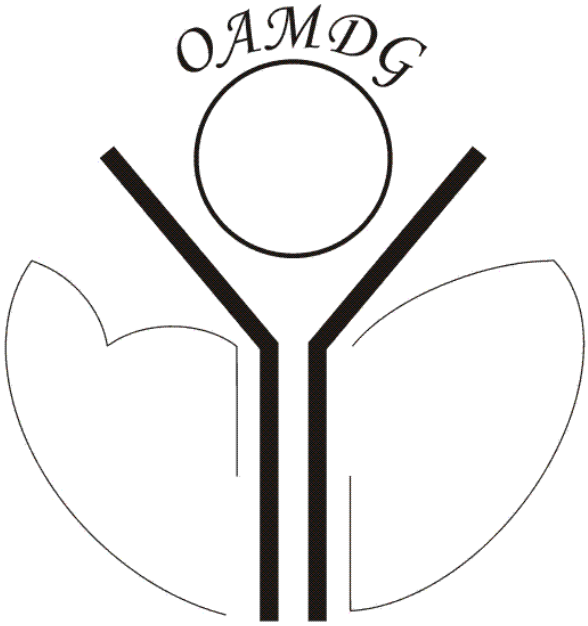
Bratislava, August 2011 **Matúš Jókay**

Ďakujem...

(nielen) za

trpezlivosť,
ústretovosť,
a príjemné pracovné prostredie

svojmu Vedúcemu!



Abstrakt

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program: Aplikovaná informatika
Autor: Matúš Jókay
Názov dizertačnej práce: Steganografia v obraze a videu
Školiteľ: Prof. RNDr. Otokar Grošek, PhD.

August, 2011

Práca sa zaoberá steganografiou v oblasti statického obrazu a videa. Po objasnení základných pojmov steganografie sa venuje pozornosť návrhu a analýze štyroch steganografických systémov založených na: nekomprimovanom obrazovom súbore BMP, nekomprimovanom videu formátu RAW AVI, komprimovanom obrazovom súbore JPEG a komprimovanom videu uloženom v multimediálnom kontajneri MP4. Steganografické systémy implementované na nekomprimovaných údajoch využívajú na vkladanie metódu modifikácie najmenej významných bitov v priestorovej oblasti použitého farebného modelu. Systém navrhnutý pre komprimačný algoritmus JPEG využíva techniku modifikácie najmenej významných bitov DCT koeficientov v spojení s Huffmanovým kódovaním. Napokon systém založený na súborovom formáte využíva internú štruktúru GOP videa kódovaného štandardmi rodiny MPEG a spôsob fyzického uloženia údajov v kontajneri MP4. V záverečnej časti práce sa prezentujú experimenty v oblasti vyrovnávania sledovaných štatistík nosiča, do ktorého sa vkladá správa. Pri optimalizácii sa porovnávajú genetické algoritmy, neurónové siete a horolezecký algoritmus.

Kľúčové slová: steganografia, kódovanie, obraz, video, BMP, JPEG, RAW-AVI, MP4

Abstrakt

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTA OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Degree course:	Applied informatics
Author:	Matúš Jókay
Thesis:	Steganography in the images and video
Supervisor:	Prof. RNDr. Otokar Grošek, PhD.

August, 2011

This thesis deals with the steganography in the area of static images and video. After explanation of the basic notions in steganography, the attention is focused on the design and analysis of the four steganographic systems based on:

- uncompressed BMP image file,
- uncompressed RAW AVI video file,
- compressed JPEG image file, and
- compressed video stored in the MP4 multimedia container.

Systems based on uncompressed data use the least significant bit (LSB) modification method in the spatial domain of the used color model. The system designed for the JPEG compression algorithm uses the modification of the least significant bits of the DCT coefficients in connection with the Huffman encoding. The system based on the MP4 file format uses the internal structure GOP of the video encoded by the MPEG family standards. Moreover, it modifies the structure of the data storage in the MP4 container. The last part of the thesis contains experimental results. Our goal was to balance the statistics of the carrier affected by the embedded hidden message. The comparison is provided for the genetic algorithms, neural networks, and the hill-climbing algorithm, respectively.

Keywords: steganography, coding, image, video, BMP, JPEG, RAW-AVI, MP4

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam skratiek a značiek

AC	označenie všetkých DCT koeficientov okrem koeficientu DC
AVI	multimediálny súborový formát, z angl. *Audio-Video Interleave*
BMP	bitmapový grafický súborový formát
DC	označenie DCT koeficientu zodpovedajúceho nulovej frekvencii
DCT	diskrétna kosínusová transformácia, z angl. *Discrete Cosine Transform*
DFT	diskrétna Fourierová transformácia, z angl. *Discrete Fourier Transform*
DSSS	metóda transformácie nosného signálu, z angl. *Direct Sequence Spread Spectrum*
DV	označenie digitálneho videa, z angl. *Digital Video*
EXIF	formát na uchovávanie údajov vo fotografiách, z angl. *EXchangeable Image File*
EOF	značka konca súboru, z angl. *End Of File*
EOI	značka konca obrazových údajov, z angl. *End Of Image*
ET	označenie schémy prahovej entropie
F5	steganografický algoritmus
FAT	bitmapový súborový systém, z angl. *File Allocation Table*
GOP	skupina snímkov medzi dvoma I snímkami, z angl. *Group Of Pictures*
H.261	video štandard určený na prenos v dátovom toku ISDN
HPHB	množina špecificky zvolených údajov, z angl. *High Precision Histogram Bin*
ISDN	digitálna telefónna služba na prenos dát
JPEG	označenie stratového komprimačného štandardu
LPHB	množina obsahujúca aspoň 2 HPHB, z angl. *Low Precision Histogram Bin*
LSB	označenie najmenej významného bitu, z angl. *Least Significant Bit*
MP4	multimediálny súborový formát
MPEG-1	video štandard určený na prenos v dátovom toku PAL a NTSC
MPEG-2	video štandard určený na prenos v dátovom toku od 4 do 9 Mbit/s
NTSC	televízna norma (30 snímkov za sekundu)
PAL	televízna norma (25 snímkov za sekundu)
RA	trieda samoopravných kódov
RGB	aditívny farebný model
RLE	technika bezstratovej kompresie, z angl. *Run-Length Encoding*
QF	faktor kvality JPEG kompresie
SEC	označenie schémy selektívneho vkladania
TCP/IP	rodina sieťových protokolov
VLC	kód s premenlivou dĺžkou, z angl. *Variable Length Code*
YCbCr	farebný model s oddelenou jasovou zložkou

Úvod

Snaha o ukrytie existencie informácie sprevádza človeka počas celej jeho histórie jestvovania. Prvé písomné zmienky o technikách ukrývania jestvovania správ pochádzajú zo starého Egypta spred štyroch tisícročí [47]. Od čias Egypťanov sa vynašlo mnoho steganografických techník a oblastí ich využitia.

Podobne ako kryptografii, ani steganografii nebol až do minulého storočia venovaný dostatočný priestor medzi vednými disciplínami. Ukrývanie jestvovania informácií bolo považované skôr za istý druh umenia. Avšak neodškriepiteľné historické fakty pripomínajú, ako toto umenie zasahovalo do priebehu ľudských dejín v mocensko–politickom utváraní spoločnosti. Nakoľko technické vymoženosti minulých storočí umožňovali realizáciu steganografických techník iba veľmi dobre financovaným organizáciám, nebolo možné do oblasti výskumu steganografických systémov zapojiť veľké množstvo ľudí. Aj týmto sa dá vysvetliť malý záujem vedeckého výskumu venovaný tejto oblasti.

Avšak s príchodom rozvoja informatiky a digitalizáciou informácií sa oblasť využitia steganografie natoľko rozšírila, že sa dostala do popredia záujmu odbornej verejnosti. Hoci samotné ukrývanie jestvovania správ je možné v reálnom svete realizovať oveľa väčším množstvom spôsobov než v tom digitálnom, hlavným nedostatkom steganografie reálneho sveta ostáva doručenie správy k príjemcovi. Problémom je nielen neprijateľné časové zdržanie, ale aj samotný transport správy. Tieto dva základné nedostatky odstránil digitálny prenos informácie. Pomocou neho je možné správu prenášať v rámci celej planéty v reálnom čase.

Globalizácia umožňuje komunikáciu ľudí celého sveta bez časového a priestorového obmedzenia. Počet osôb, s ktorými môže každý človek komunikovať, sa rádovo zvýšil oproti technickým možnostiam komunikácie predošlých storočí. Zavedenie celosvetovej siete informácií dovoľuje zdieľať textové, zvukové a obrazové dokumenty všetkými účastníkmi siete. Takto sa Internet a jeho všeobecná dostupnosť stali vhodným komunikačným kanálom na šírenie steganografických správ.

Techniky steganografie sa využívajú v zásade dvoma skupinami ľudí. Prvou sú tí, ktorí pomocou ukrývania dodatočných informácií do média chcú zabezpečiť presné určenie páchateľa pri zneužití aktíva, ktoré je chránené steganografickými informáciami. Špeciálnou oblasťou tohto typu steganografie je tzv. vodotlač — steganografické vkladanie rôznych informácií do digitálnych médií (napríklad do obrázkov, zvuku, videa, dokumentov). Ľudia, ktorí zneužívajú možnosti steganografie, patria do druhej skupiny. Zneužitím chápeme využitie steganografických systémov na dosahovanie cieľov, ktoré sú v rozpore so zákonom. Príkladom môže byť dorozumievanie členov teroristických skupín ohľadom plánovaných útokov.

Pre zníženie možnosti zneužitia systémov na ukrývanie existencie informácií je nutné venovať pozornosť možnostiam detekcie utajených správ. Touto oblasťou sa zaoberá stegoanalýza. Vzhľadom na šírku komunikačného kanála Internetu je dôležité, aby bola detekcia optimálna (jej kritériom je rýchlosť a efektívnosť). Žiaľ, skupina steganografických systémov založených na tom istom médiu zväčša vyžaduje svoj vlastný steganoanalytický prístup. To znemožňuje jednoduché zjednotenie steganoanalytických metód. Navyše, mnohé steganoanalytické postupy sú výsledkom skôr umenia než vedy, nakoľko pri voľbe parametrov steganografických algoritmov nie je možné exaktne určiť ich počiatočné hodnoty. Inicializácie (a niekedy i vyhodnocovanie výstupov) algoritmov sú tak dané skúsenosťami toho, kto vykonáva steganoanalýzu.

Stále ostáva otvorená otázka návrhu praktického (nielen teoretického) optimálneho steganografického systému (v tomto prípade sú kritériami optimálnosti odolnosť voči narušeniu ukrytej správy a odolnosť voči steganoanalýze).

Steganografické systémy i steganoanalýza týchto systémov vyžadujú použitie istých prvkov (napríklad vrstvy systému alebo rôzne metriky efektivity), ktoré je možné zovšeobecniť a používať jednotne pre všetky. Podobne jestvuje množina elementov, ktoré síce nie sú použiteľné všeobecne, ale u ktorých je možné definovať spoločné komunikačné rozhranie. Problémom je skutočnosť, že zatiaľ nejestvuje definovaný štandard steganografických primitív, a dokonca ani návrh zoznamu nejakých základných jednotiek.

Štruktúra tejto práce je nasledovná. V prvých troch kapitolách sa snažíme v krátkosti objasniť súčasný stav problematiky na poli steganografie. V prvej rozoberáme základné pojmy steganografie, v druhej diskutujeme obrazovú steganografiu a v tretej video steganografiu.

V ďalších kapitolách podávame súhrnný pohľad na oblasti, v ktorých sme počas výskumu pracovali: v kapitolách 4 a 6 sa jedná o implementáciu dvoch steganografických systémov v oblasti statického obrazu a v kapitolách 5 a 7 dvoch systémov v oblasti videa. V každej z uvedených oblastí (obraz a video) sa jeden systém venuje nekomprimovaným objektom, kým druhý je zameraný na komprimované formy obrazu a videa. Okrem návrhu jednotlivých steganografických systémov načrtávame možnosti ich stegoanalýzy.

V kapitole 8 prezentujeme výskum v oblasti optimalizácie steganografických systémov vzhľadom na sledované štatistiky. Porovnávame efektivitu viacerých algoritmov z oblasti evolučných výpočtov a neurónových sietí.

Kapitola 1

Steganografia vo všeobecnosti

1.1 Úvodné pojmy

Steganografia je veda, ktorá sa zaoberá návrhom, analýzou a implementáciou algoritmov, metód a systémov, ktoré umožňujú:

- prenášať informáciu nepodozrivým komunikačným kanálom,
- utajiť samotné jestvovanie správy v komunikačnom kanále.

Steganoanalýza (alebo tiež *stegoanalýza*) je vedná disciplína, ktorá sa zaoberá návrhom, analýzou a implementáciou algoritmov, metód a systémov, ktoré umožňujú potvrdiť alebo vyvrátiť existenciu informácie v nevinne vyzerajúcom komunikačnom kanále. Ako bude ukázané neskôr, primárnym cieľom nie je informáciu extrahovať.

V gréčtine slovo *stegos* znamená strechu (ukrytie), *graphos* znamená zobrazovanie, písanie. Spolu je to teda ukrývanie písma alebo obrazov [21].

Spočiatku sa nerobil rozdiel medzi pojmami kryptografia a steganografia. Ioannus Trithemius vo svojom spise, ktorý začal písať v roku 1499 (a nikdy ho žiaľ nedokončil), prvý raz používa pojem steganografia a používa ho vo význame oboch súčasných jazykových výrazov (kryptografia a steganografia) [78]. Podobne sa vyjadruje vo svojej knihe z roku 1665 Caspar Schott, pričom táto kniha obsahovala už popis viacerých kryptosystémov [71]. Jedným z prvých, ktorí odlíšili pojem steganografie od kryptografie, bol John Wilkins, zakladateľ spoločnosti Royal. Steganografiu definoval ako tajné písmo [5]. Vlastným vedným odborom (odvodeným z teoretickej informatiky a matematiky) sa stala v poslednom desaťročí minulého tisícročia. Od 90–tich rokov minulého storočia sa vydávajú publikácie a organizujú konferencie zamerané iba na oblasť steganografie a jej aplikácií.

Ako vidno z definície, steganografia využíva metódy utajenia komunikácie s cieľom realizovať utajený (skrytý) prenos informácie, ktorý prebieha v pozadí neutajenej komunikácie. Medzi steganografiou a kryptografiou je podstatný rozdiel práve v tom, že kryptografia utajuje iba informačný obsah, nie samotnú existenciu správy. Zjednodušene povedané, kryptografia transformuje pôvodne čitateľnú správu na takú, ktorú dokáže spätne transformovať na čitateľnú iba legitímny príjemca. Transformovaná správa sa posiela verejným komunikačným

kanálom, kde k nej môže mať prístup aj nelegitímny príjemca. Ten sa môže pokúsiť o transformáciu správy do čitateľnej podoby, ale jeho úspešnosť bude závisieť od odolnosti kryptografického systému, ktorý bol na transformáciu pôvodnej správy použitý.

Steganografia študuje rôzne techniky ukrývania existencie nejakej (druhotnej) správy v inej (prvotnej) správe. Primárna správa sa nazýva *nosič* alebo *obálka*, sekundárna sa označuje prívlastkom *tajná*. Nosič informácie, v ktorom je ukrytá tajná správa, sa nazýva *steganografické médium*.

Cieľom steganografie je zakryť skutočnosť, že sa prenáša niečo tajné. Jej podstatou je snaha ukryť správu tam, kde ju nikto nebude hľadať. Keď sa totiž prenášajú údaje, ktoré vyzerajú podozrivo[1], je jednoduché komunikáciu narušiť, pozmeniť, alebo úplne zničiť. Ak však prenášané údaje vyzerajú nevinne, je oveľa ťažšie rozhodnúť, či sa na pozadí takéhoto prenosu prenáša nejaká tajná informácia alebo nie. Tajná správa sa teda prenáša v inej správe. Narušením steganografického systému sa myslí skutočnosť odhalenia existencie prenášania správy v nejakej inej správe. Nie je potrebné, aby sa dala ukrytá správa z nosiča extrahovať. Na narušenie stačí, že sa o jej jestvovaní vie.

Podľa [15] steganografia pokrýva tri ciele bezpečnosti: dôvernosť, schopnosť prežitia (robustnosť, odolnosť) a neodhaliteľnosť (nedetegovateľnosť).

Dôvernosť Dôvernosť je jednou z troch požiadaviek informačnej bezpečnosti (tými sú integrita, dostupnosť, dôvernosť). Pojmom dôvernosť označujeme schopnosť uchovať tajomstvo v tom zmysle, že neautorizovaná osoba nemôže získať prístup k tajnej informácii. Vlastnosť dôvernosti je jadrom steganografie. Steganografia pristupuje k požiadavke dôvernosti odlišným spôsobom ako kryptografia. V prípade kryptografie neautorizovaná osoba vidí údaje, ktoré obsahujú tajnú informáciu, ale nedokáže ju sprístupniť. Môže sa však pokúšať narušiť kryptografický systém. V prípade steganografie sa však navyše utajuje samotné ukrytie informácie, takže neautorizovaná osoba nemá ani poňatie o prenose tajnej informácie. Je zrejmé, že vzhľadom na oblasť dôvernosti je informácia v steganografii chránená kvalitatívne lepšie než v kryptografii.

Odolnosť voči narušeniu Schopnosť prežitia sa týka prenosu tajnej informácie komunikačným kanálom. Vo všeobecnosti sa v steganografii tajná informácia posiela nezabezpečeným komunikačným kanálom. Utajenie sa dosahuje pomocou splnenia vyššie opísanej požiadavky dôvernosti (utajuje sa samotná skutočnosť prenosu informácie). Prenos správy pozostáva z odoslania správy, prechodu správy komunikačným kanálom a jej prijatia legitímnym príjemcom. Schopnosť prežitia charakterizuje úspešnosť celého procesu prenosu správy bez poškodenia tajnej informácie. Preto ďalším cieľom steganografie je vyvíjať také algoritmy ukrývania informácie do nosiča, ktoré zabezpečia integritu tajnej informácie aj v prípade modifikácie nosiča správy počas jeho transportu komunikačným kanálom. Odolnosť voči zmenám nosiča úzko súvisí s množstvom informácie, ktorú je možné do nosiča ukryť. Čím väčšiu odolnosť požadujeme, tým menej informácie môžeme v nosiči ukryť. Preto je dôležité vopred správne odhadnúť vlastnosť prenosového kanála v zmysle schopnosti správy v ňom prežiť. Podľa ohodnotenia tejto vlastnosti potom môžeme nastaviť správny pomer medzi odolnosťou a priestorom na uloženie informácie v nosiči (tzv. kapacita nosiča).

[1] napríklad prenos šifrovanej elektronickej pošty môže vyvolávať pochybnosti, pokiaľ sa jedná o komunikáciu medzi osobami podozrivými z teroristickej činnosti

Neodhaliteľnosť Nemá žiaden význam ukrývať informácie, ak nelegitímny príjemca vie, ako zistiť, či sa nejaká tajná informácia v nosiči ukrýva. Ešte horšia situácia nastáva v prípade, že útočník dokáže túto správu z nosiča získať a prečítať. Podmienka nedetegovateľnosti v sebe teda zahŕňa dve skutočnosti. Prvou je nemožnosť rozlíšenia, či nosič obsahuje skrytú správu, iba na základe vlastností nosiča. Druhou je nemožnosť potvrdenia (vyvrátenia) existencie tajnej správy v médiu v prípade, že je útočníkovi známa technika vkladania tajnej správy do nosiča. Preto musí byť steganografický algoritmus navrhnutý tak, aby nijakým spôsobom výrazne nemenil tie vlastnosti nosiča, ktoré je možné použiť na odlíšenie steganografického média od čistého. Navyše, na základe známeho algoritmu nesmie byť možné určiť, či nosič obsahuje tajnú správu alebo nie. Skúsenosť a história ukazujú, že je vhodné, ak sa aj v prípade steganografie zachováva Kerckhoffov princíp aplikovaný na túto doménu. To znamená, že dosahovanie nedetegovateľnosti (bezpečnosti steganografického systému) nesmie byť závislé od utajenia techniky ukrývania informácie do média. Preto sa v súčasnosti väčšina steganografických systémov navrhuje s tajným parametrom (kľúčom), bez ktorého znalosti nie je možné správu nielen extrahovať, ale ani detegovať jej existenciu v nosiči.

Vzhľadom na cieľ steganografie (ukrývanie informácie v nepodozrivo vyzerajúcom médiu spôsobom, ktorý znemožní nelegitímnemu príjemcovi zistiť alebo zničiť prítomnosť tajnej správy v nosiči) je možné definovať niekoľko kritérií merania efektivity steganografických techník:

- Množstvo údajov, ktoré je možné do média vložiť,
- zložitosť detekcie ukrytej správy a
- zložitosť zničenia tajnej správy.

Množstvo informácií, ktoré je možné do nosiča vložiť (označované tiež ako *kapacita média*), je ukazovateľom efektivity v tom zmysle, že čím je toto množstvo väčšie, tým lepšia je steganografická technika.

Ťažkosť detekcie sa vzťahuje na skutočnosť, nakoľko je zložité zistiť, či sa v médiu nachádza ukrytá správa. Medzi kapacitou nosiča a zložitosťou detekcie správy je zvyčajne nepriamo úmerný vzťah: zvyšovaním kapacity sa znižuje zložitosť detekcie (zväčšuje sa možnosť odhalenia ukrytej správy), pretože vkladanie správy v oveľa väčšej miere modifikuje vlastnosti pôvodného média bez vloženej správy.

Zložitosť zničenia tajnej informácie sa zakladá na skutočnosti, že útočník, ktorý odchytí steganografické médium, by nemal byť schopný jednoducho odstrániť tajnú informáciu z média. Pojmom „jednoducho“ sa chápe v optimálnom prípade to, že útočník pre odstránenie správy musí poškodiť aj samotný nosič informácie.

Delenie steganografických techník je možné robiť podľa mnohých kritérií. Jedno z možných delení je podľa nosiča tajnej informácie; iné podľa toho, akým spôsobom sú informácie ukrývané.

Delenie podľa typu média

Delenie steganografických techník podľa nosiča informácie je založené na rôznorodosti médií, ktoré je možné použiť na ukrytie tajnej informácie. Vlastnosti rôznych nosičov určujú, akým spôsobom sa tajná informácia vkladá.

Napríklad mnohé techniky, ktoré pracujú s grafickými súbormi, nahrádzajú najmenej významný bit obrazovej informácie každého bodu obrazu jedným bitom tajnej informácie. Aj keď sa výber bitov, ktoré majú byť nahradené, v jednotlivých algoritmoch odlišuje, myšlienka pre celú túto rodinu techník je rovnaká. To znamená, že na základe znalosti typu údajov, ktoré sú nosičmi tajnej informácie, vieme povedať, na akom mieste môžu byť údaje ukryté. To, akým spôsobom bude tajná informácia vložená, jednotlivé techniky daného typu odlišuje.

Iným príkladom steganografického média je komunikačný sieťový protokol IP verzie 4. Niektoré polia hlavičky je možné použiť na ukrytie prenášanej tajnej informácie. Viac podrobností o technikách sieťovej steganografie sa nachádza v časti 1.3 na strane 18.

Najčastejšie sa v oblasti steganografie ako nosiče využívajú:

- Nevyužité polia hlavičiek rôznych súborových formátov,
- oblasti súborov, ktoré aplikácie nekontrolujú pri spracovaní obsahu súboru (metadáta, formátovanie, údaje uložené za značkou konca súboru),
- redundantné informácie (súbory s multimediálnym obsahom — zvukové, obrazové, animácie).

Delenie podľa spôsobu ukrývania údajov

Pôvodne sa metódy ukrývania informácie do steganografického média delili na tri skupiny techník: techniky založené na vkladaní, algoritmické a gramatické techniky.

Techniky založené na vkladaní pracujú na princípe vkladania tajnej informácie do cieľového nosiča. Špecifickou vlastnosťou týchto metód je rovnaké miesto vkladania údajov pri každom ich použití. Algoritmy tejto skupiny pracujú tak, že hľadajú vhodné miesta v médiu, ktoré môžu prepísať vkladanou tajnou informáciou. Vhodné miesta sú také, ktorých modifikácia nemá výrazný vplyv na kvalitnú interpretáciu údajov nosiča. Môžu to byť napríklad nepoužívané polia hlavičiek rôznych formátov súborov, pri farebných obrázkoch to môžu byť farebné tabuľky, redundantné údaje o farbe, alebo vo zvukových súboroch nadbytočné zvukové údaje... Pretože tajná informácia sa umiestňuje vždy na to isté miesto, označenie skupiny týchto steganografických techník je odvodený od vkladania.

Algoritmické steganografické techniky využívajú rôzne postupy, ako vybrať optimálne miesto v nosiči pre vloženie tajnej informácie. Tieto metódy sú menej náchylné na poškodenie údajov nosiča, nakoľko nevkladajú údaje tajnej správy vždy na to isté miesto bez analýzy vhodnosti jej umiestnenia. Špeciálnu skupinu uvedených techník tvoria také, pri ktorých je umiestnenie tajnej správy určované na základe nejakého (nie nutne tajného) parametra. Algoritmus metódy takéhoto typu musí byť natoľko sofistikovaný, aby žiadna z možných hodnôt parametra umiestnenia správy neviedla k zlému výberu miesta modifikácie údajov nosiča. Pod zlým miestom treba rozumieť takú lokalitu, ktorej zmena spôsobí pozorovateľné odchýlky od bežných vlastností média.

Techniky založené na vkladaní alebo algoritmoch nejakým spôsobom vkladajú tajnú správu do nosiča. Na rozdiel od nich, gramatické techniky steganografie nevyžadujú žiadne médium, do ktorého by sa správa vkladala, pretože samé podľa tajnej správy generujú jej nosič. Táto trieda techník používa tajnú správu na generovanie média v tom zmysle, že podľa nej a na základe pravidiel nejakej gramatiky vytvorí súbor nosiča (ktorý je zároveň médiom). Výhodou týchto techník je, že znemožňujú štatistické metódy hľadania skrytých správ. Aplikácie, ktoré analyzujú jestvovanie tajnej správy v médiu na základe štatistických vlastností nenájdu v takto vytvorenom nosiči žiadne odchýlky od bežných atribútov.

Delenie podľa metódy ukrývania tajnej informácie

Staršie delenie začleňovalo jednotlivé techniky podľa toho, akým spôsobom sa tajná správa začleňovala do nosiča. Toto novšie členenie techník steganografie zahŕňa do hodnotenie nielen odpoveď na otázku „ako“, ale aj „kde“ sa integrovanie tajnej informácie s nosičom deje. V súčasnosti je viac používané. Abstrahuje od vlastností média a uľahčuje analýzu efektívnosti a korektnosti skúmanej steganografickej techniky. V podstate jestvujú tri spôsoby ukrývania informácií: vkladanie, nahradzovanie a generovanie.

Pri metódach **vkladania** je podstatné nájsť v nosičoch také oblasti, ktoré aplikácia navrhnutá pre prácu s nosičom povinne nevyužíva. Do takýchto ignorovaných oblastí je možné vložiť tajnú správu bez obavy, že bude zničená. Napríklad mnohé súbory vo svojej internej štruktúre údajov využívajú symbol `EOF` (*End Of File*) na označenie konca úseku údajov potrebných pre aplikáciu. Aplikácia po prečítaní tohto znaku ďalšie údaje zo súboru nespracúva. Preto je možné vložiť tajnú správu za tento symbol.

Substitučné metódy vyhľadávajú najmenej významné informácie nosiča, a tieto nahrádzajú informáciami, ktoré je žiadané utajiť. Nevýhodou týchto metód je malá robustnosť voči zmenám, ktoré na nosiči môže konať legitímna aplikácia. Takouto metódou je napríklad vyššie spomínaná technika zmeny najmenej významného bitu v obrazových súboroch. Substitučné metódy sú úspešné vďaka neschopnosti ľudských zmyslov postrehnúť malú zmenu údajov nosiča.

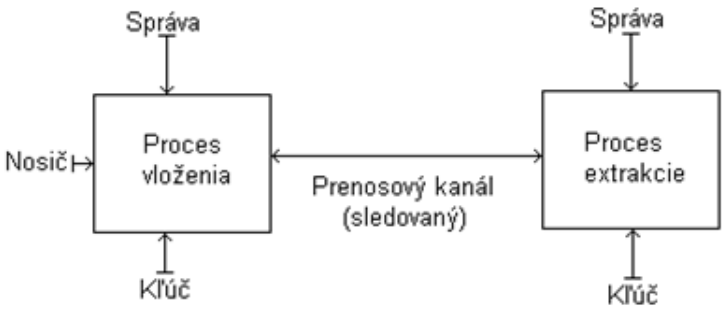
Metódy **generovania** sú úplne odlišné od predošlých dvoch spomínaných. Kým v predchádzajúcich sa využíval jestvujúci nosič na vloženie informácie, pri tejto metóde sa žiadne médium neupravuje. Nosič tajnej informácie sa generuje priamo podľa potreby — na základe informácie, ktorú je potrebné ukryť. Metódu generovania využívajú predovšetkým syntaktické a sémantické techniky steganografie.

Využitie steganografie v súčasnosti

V súčasnosti sa kladie dôraz na výskum v oblasti digitálnej steganografie. V oblasti digitálnej techniky jestvuje veľa technológií, ktoré umožňujú implementáciu steganografických systémov. Patria sem predovšetkým textové súbory, statické obrázky, pohyblivý obraz a zvuk. Množstvo nástrojov, ktoré dokážu manipulovať s vymenovanými médiami za účelom steganografického prenosu informácie možno nájsť v rámci celosvetovej pavučiny stránok[2].

Veľmi špecifickým prípadom využitia techník ukrývania informácií je oblasť *digitálnej vodotlače*, tzv. *watermarking*. Pomocou rôznych techník možno „označiť“ obrázky, hudbu či video digitálnou značkou. Dôležitou vlastnosťou tejto „značky“ je to, že proces jej tvorby veľmi úzko súvisí s médiom, ktoré je označované. Tým sa líši od steganografie: závislosťou ukrytej informácie od samotného nosiča. Navyše, v oblasti digitálnej vodotlače je kladený najväčší dôraz na odolnosť vodoznaku voči narušeniu. V dôsledku veľkej robustnosti je množstvo tajnej informácie, ktoré možno do média vložiť, oveľa menšie, než pri klasických steganografických technikách. Väčšia robustnosť sa dosahuje vhodnou voľbou kódovania tajnej informácie do nosiča, čo má za následok modifikáciu nosiča v oveľa väčšej miere, ako pri steganografických technikách. Voľne dostupné nástroje na testovanie robustnosti vodoznakov sú napríklad StirMark a unZign [62].

[2]Napríklad stránka `www.stegoarchive.com` ponúka rozličné nástroje pre rôzne platformy: DOS, Windows, Java, Macintosh, OS/2, Unix, OpenBSD, Linux, Amiga.



Obr. 1.1: Schéma základného steganografického systému.

Zaujímavosťou je skutočnosť, že digitálna vodotlač využíva aj nosiče, ktoré nie sú digitálne. Napríklad firma Xerox vyvinula systém vodotlače pre tlačené obrázky, ktorý pomocou kľúčového obrázka (tzv. vzoru) generuje označený obraz. Keď je takto označený obraz s istým posunutím preložený cez pôvodný obrázok, zviditeľní sa kľúčový vzor. Pomocou tejto techniky je možné dosiahnuť veľmi vysokú mieru robustnosti — až do tej miery, že jedine rozsiahle zničenie obrazu znemožní zviditeľnenie kľúčového obrazu [73].

S rozvojom digitálnej vodotlače sa tiež vyvíjali rôzne formy útokov. Doteraz známe útoky možno (podľa [61]) rozdeliť do nasledovných skupín: pridávanie šumu, filtrovanie, orezávanie, kompresia, rotácia a zmena mierky, štatistické priemerovanie, viacnásobné aplikovanie vodotlače.

Prehľad celej oblasti vodotlače a špeciálny návrh systému odolného voči afinným transformáciám je v dizertačnej práci Radovana Ridzoňa [65].

Steganografiu je možné v súčasnosti nájsť v mnohých ďalších oblastiach (technológia programovateľných infračervených ovládačov, špeciálne sklá prepúšťajúce len istú vlnovú dĺžku, neviditeľný atrament, ukrývanie správ pomocou DNA, žargónový slovník, prázdne oblasti na dátových nosičoch...), nielen v obrazových súboroch na Internete.

1.2 Definícia steganografického systému

Zvyčajne sa v literatúre uvádza ako vôbec prvá (hoci neformálna) definícia steganografického systému schéma s názvom *väzenský problém*. Nech väzni Alica a Bob sú pod stálym dohľadom dozorkyne Evy. Akákoľvek komunikácia medzi väzňami prechádza cez Evu, takže táto môže komunikáciu buď povoliť alebo zahodiť[3] v prípade, že sa jej javí podozrivá. Cieľom väzňov je komunikovať takým spôsobom, aby sa to Eve nezdalo podozrivé. Preto sa snažia pri prenose nejakej informácie použiť taký komunikačný prostriedok, ktorý nebudí podozrenie. Obrázok číslo 1.1 znázorňuje schému jednoduchého steganografického systému, ktorý môžu Bob a Alica používať.

Jadro steganografického systému je definované použitou steganografickou schémou. Steganografická schéma je matematický model, ktorý je možné skúmať a analyzovať metódami matematiky a informatiky. Jej konkrétna implementácia sa nazýva steganografickým systémom.

[3] alebo tiež pozmeniť

Definícia 1. ***Steganografická schéma*** *je určená nasledovnými dvoma zobrazeniami:*

$$\mathrm{e} : \mathscr{C} \times \mathscr{K} \times \mathscr{M} \to \mathscr{C},$$

$$\mathrm{d} : \mathscr{C} \times \mathscr{K} \to \mathscr{M},$$

kde $\mathscr{C}$ je množina nosičov (médií), $\mathscr{K}$ množina kľúčov, $\mathscr{M}$ množina správ a $s = \mathrm{e}(c, k, m)$, $s \in \mathscr{C}$ sa nazýva steganografické médium s nasledovnými vlastnosťami:

$$\mathrm{d}(\mathrm{e}(c, k, m), k) = m, c \in \mathscr{C}, k \in \mathscr{K}, m \in \mathscr{M},$$

$$\mathrm{d}(\mathrm{e}(c, k_1, m), k_1) \neq \mathrm{d}(\mathrm{e}(c, k_2, m), k_2), k_1, k_2 \in \mathscr{K} : k_1 \neq k_2, c \in \mathscr{C}, m \in \mathscr{M}.$$

Navyše, v prípade symetrickej steganografickej schémy musí byť splnená vlastnosť

$$\mathrm{d}(\mathrm{e}(c, k_1, m), k_2) \neq m, k_1, k_2 \in \mathscr{K} : k_1 \neq k_2, c \in \mathscr{C}, m \in \mathscr{M}.$$

Poznámka 1. *V našej práci uvažujeme iba symetrické steganografické schémy.*

Definícia 2. ***Kapacita média*** *je definovaná vzťahom $\boldsymbol{cap} = |N|$ a vyjadruje sa v bitoch. N je najdlhšia správa, ktorú je možné vložiť do nosiča a operátor $|N|$ vracia dĺžku správy, danú počtom symbolov v $\mathbf{Z}_2$, ktoré ju reprezentujú.*

Poznámka 2. *Kapacita média vyjadrená nami definovaným spôsobom vyžaduje možnosť určenia najdlhšej správy, ktorú možno vložiť do nosiča. Uvedenú definíciu sme uviedli v tejto forme kvôli jednoduchosti vyjadrenia efektivity procesu vkladania (viď definícia ďalej). V praxi často nie je možné priamo určiť kapacitu steganografického systému pre všetky médiá, pretože množstvo správy, ktorú možno vložiť, veľmi úzko závisí od použitého algoritmu a vlastností každého nosiča zvlášť.*

Definícia 3. ***Dopad vloženia správy*** *na médium vyjadruje miera nezhody nosiča so steganografickým médiom $\mathrm{D}(c, s)$. D je funkcia merania odchýlky steganografického média od čistého nosiča. Najčastejšie sa na meranie odchýlky používa stredná kvadratická chyba.*

Definícia 4. ***Efektivita procesu vkladania*** *správy do nosiča je definovaná vzťahom $\boldsymbol{cap}/\mathrm{D}(c, s)$ a vyjadruje priemerný počet bitov (ktoré je možné vložiť do média) na jednotku dopadu vloženia správy.*

Základný steganografický systém musí pozostávať minimálne z nosiča informácie, procesov (funkcií) vloženia a extrakcie a prenosového kanála. V rozšírení je možné použiť pre proces vkladania a rekonštrukcie správy aj kľúčovanie. Prípadné modifikácie schémy, ktoré uvažujú prítomnosť informácie o originálnom (nemodifikovanom) médiu pri procese kódovania a dekódovania tajnej informácie, je možné integrovať so schémou z obrázka č. 1.1[4].

V nasledujúcich častiach venujeme bližšiu pozornosť základným elementom steganografickej schémy: prenosovému kanálu, nosiču a kódovacím funkciám tajnej informácie.

[4] v tomto prípade uvažujeme dodatočnú informáciu ako interný parameter procesov vkladania či extrakcie správy

Prenosový kanál

V schéme steganografického systému modelujeme prenosový kanál ako otvorený voči pozorovateľovi („dozorcovi“) komunikácie medzi dvoma účastníkmi systému. Navyše, pridávame vlastnosť, že pozorovateľ môže do komunikácie zasahovať: môže ju zadržať, prepustiť, alebo pozmeniť. Potom môžeme definovať nasledovné tri typy dozorcov komunikácie:

pasívny: Jedná sa o pozorovateľa v pravom slova zmysle, pretože nijako nemodifikuje správu pred jej doručením legitímnemu príjemcovi steganografického systému. Správa je doručená iba v prípade, ak test pozorovateľa na prítomnosť tajnej správy v prenose je negatívny. V opačnom prípade správa nebude doručená.

aktívny: V tomto modeli „dozorca“ komunikácie modifikuje akúkoľvek správu posielanú medzi účastníkmi steganografickej schémy; zmenu uskutočňuje aj vtedy, keď test prítomnosti ukrytej správy je negatívny.

zlomyseľný: Pozorovateľ sa snaží zachytiť tajnú komunikáciu účastníkov systému. Na to využíva špecifické vlastnosti steganografickej schémy. Tento model sa používa väčšinou na analýzu bezpečnosti známeho steganografického systému. Pri známom systéme (známy prenosový kanál, nosič, funkcie vkladania a extrakcie tajnej správy) a neznámom kľúči sa pozorovateľ snaží zachytiť a odhadnúť obsah prenášanej tajnej správy. Niektorí autori definujú zlomyseľného pozorovateľa v odlišnom kontexte: že je známy aj steganografický kľúč, ale prenášaná tajná správa je navyše zašifrovaná asymetrickým kryptografickým systémom. Cieľom útoku je získanie a dešifrovanie komunikácie (napríklad pomocou útoku „muž v strede“).

Väčšina steganografických schém modeluje prvý typ pozorovateľa. V takom modeli nie je potrebné implementovať mechanizmy pre opravu vložených tajných údajov, ako je tomu v prípade druhého modelu.

Výber steganografického média

Ako bolo spomínané v časti Využitie steganografie v súčasnosti na strane 7, na rozdiel od vodoznakov, tajná informácia v steganografickom systéme nemusí byť spätá s nosičom tejto informácie. To zaručuje steganografii väčšiu voľnosť voľby média, do ktorého sa správa ukryje, ako je tomu pri vodoznakoch. Fantázia výberu býva obmedzená dostupnosťou média a vlastnosťami prenosového kanála[5].

Kvôli lepšiemu (z hľadiska detekcie správy) ukrytiu tajnej informácie je vhodné voliť médium, ktoré je obsahovo blízke prenášanej správe. Nie je optimálne vybrať na prenos dynamickej informácie napríklad farebne monotónny statický obraz (monotónny v zmysle obsahu veľkých plôch s rovnakou farbou). Pokiaľ to steganografický systém umožňuje, najvhodnejšie je médium generovať podľa vlastností vkladanej informácie. Optimálnosť voľby média podľa súvisu s obsahom správy súvisí s veľkosťou šumu v nosiči, ktorý sa v steganografii modeluje vložením tajnej správy do média. Pokiaľ sa zvolené médium podobá prenášanej správe, na kódovanie správy v médiu bude potrebné zmeniť menej bitov, než v prípade, keď je miera zhody oveľa menšia.

[5] Nezabúdajme, že medzi najdôležitejšie vlastnosti patrí typ pozorovateľa, popisovaný v časti Prenosový kanál na strane 10.

Pokiaľ má odosielateľ informáciu o tom, aké testy používa pozorovateľ komunikácie na rozhodovanie o prítomnosti ukrytej správy, môže zakódovať informáciu do rôznych nosičov, vykonať všetky očakávané testy vopred a následne cez verejný kanál poslať iba to médium, pre ktoré boli výsledky všetkých testov negatívne. Tiež je možné navrhnúť proces vkladania informácie do nosiča takým spôsobom, aby bola splnená podmienka negatívneho výsledku testov, ktoré sa budú na médiu počas prenosu verejným komunikačným kanálom vykonávať [46].

Steganografický kľúč

Požiadavky na kľúč v steganografii sú rovnaké ako v kryptografii. Používanie toho istého kľúča vo viacerých prenosoch tajnej správy nejakého steganografického systému (pri rovnakých parametroch systému) spôsobuje zvýšenie dodatočnej informácie o parametroch, ktoré sú od tohto kľúča odvodené. Preto sa odporúča venovať zvýšenú pozornosť pri návrhu steganografického systému aj menežmentu kľúčov.

Kľúče sa v steganografii používajú na inicializáciu rôznych parametrov a steganografických primitív systému (napríklad na inicializáciu generátora (pseudo)náhodných čísel).

Pri voľbe kľúča je nutné dbať na jeho nezávislosť od nosiča a na jeho dostatočnú silu (aby nebolo možné v reálnom čase odvodiť a/alebo prehľadať celý priestor kľúčov a kompromitovať steganografický systém).

Vlastnosti funkcií vkladania a extrahovania tajnej správy

Každá funkcia vkladania informácie je navrhnutá na základe jedného z nasledovných princípov.

Proces vkladania nijakým spôsobom nemení obsah ani reprezentáciu nosiča tajnej informácie. *Tajná správa sa kóduje odkazom na časti nosiča.*

Autor v [16] uvádza príklad tohto typu kódovacej funkcie na prenose 10 bitovej informácie medzi dvoma účastníkmi steganografického systému pomocou 1024 hudobných súborov. Napríklad počet prenesených súborov (binárny rozvoj) definuje kódovaciu funkciu. Nakoľko je prenos veľkého množstva súborov podozrivý, je možné modifikovať kódovaciu funkciu nasledovným spôsobom. Pokiaľ si obaja účastníci vopred definujú poradie skladieb, binárny rozvoj indexu prenesenej skladby je prenášanou tajnou informáciou. Nevýhodou tohto systému je to, že pozorovateľovi komunikácie sa môže zdať podozrivé prenášať viackrát tú istú skladbu medzi účastníkmi komunikácie. Ale aj to sa dá vyriešiť — napríklad nahrádzaním použitej skladby v zozname ďalšou z vopred definovaného zásobníka skladieb.

Ani v druhom prípade proces vkladania nijako nemení nosič, nakoľko ten je procesom vytváraný. *Tvorba nosiča je riadená tajnou informáciou.*

Celá oblasť lexikálnej steganografie využíva práve tento typ kódovacej funkcie. Príkladom môže byť program `SpamMimic`[6], ktorý generuje text prirodzeného jazyka obsahovo podobný najrozšírenejšiemu typu elektronickej komunikácie (90% celkového objemu) — nevyžiadanej elektronickej pošte (t.j. spamu).

Ďalším príkladom steganografického systému, ktorý využíva takúto kódovaciu funkciu, je steganografia založená na kódovej knihe. Kódová kniha konverzačných fráz bola s úspechom používaná napríklad britskou tajnou službou v druhej svetovej vojne. Každej fráze kódovej

[6] `http://www.spammimic.com`, dostupné 11. augusta 2011

knihy bolo priradené číslo, takže bolo možné vytvoriť a odoslať ľubovoľne dlhú sekvenciu čísel.

Nespornou výhodou systémov, ktoré využívajú vkladaciu funkciu na generovanie nosiča tajnej správy je kapacita prenosového kanála. Väčšinou je možné ukrývať správy ľubovoľnej dĺžky (do tej miery, aby dĺžka výsledného nosiča nebola podozrivá). Veľkosť ukrývanej správy nie je obmedzená kapacitou nosiča, ale ju definuje.

Zmena obsahu alebo reprezentácie nosiča patrí medzi najrozšírenejšie spôsoby návrhu kódovacej funkcie. Tajná informácia sa buď priamo, alebo pomocou vhodne zvolenej transformácie vkladá do média a modifikuje tak jeho obsahovú alebo reprezentačnú štruktúru.

Typ a miesto modifikácie majú najväčší vplyv na odhalenie ukrytej informácie. Čím je zmena väčšia a rozsiahlejšia, tým pozorovateľnejšia je zmena pôvodného nosiča. Preto sa všetky steganografické systémy tohto typu snažia minimalizovať zmeny nosiča pri maximalizácii objemu vloženia tajnej správy. Miesto modifikácie sa vyberá:

náhodne: Generátorom náhodných čísel sa vytvorí postupnosť, na základe ktorej sa určí, ktoré bity nosiča sa použijú na vloženie tajnej správy. Na inicializáciu náhodného generátora sa zvyčajne používa steganografický kľúč účastníkov systému. Oproti nasledovnému (sekvenčnému) typu výberu ide o zvýšenie bezpečnosti steganografického systému voči útokom, ktorých cieľom je odhalenie obsahu tajnej správy. Náhodným výberom modifikovaných miest sa však vo všeobecnosti nerieši problém vkladania šumu, na základe ktorého je možné odhaliť vloženie správy[7].

sekvenčne: Modifikujú sa miesta nosiča determinované použitým algoritmom výberu. V závislosti od typu nosiča, algoritmu vkladania a zmien sa môže jednať síce o implementačne efektívny, ale po stránke bezpečnostnej nedostatočný spôsob výberu miest, ktoré majú byť nahradené tajnou správou. Majme napríklad steganografický systém, ktorý vkladá správu do mapy obrazových bodov grafického súboru takým spôsobom, že modifikuje najmenej významný bit od začiatku mapy. Po vložení tajnej správy, ktorá zaplní iba polovicu kapacity nosiča, vypočítame napríklad frekvenčnú charakteristiku prvej aj druhej polovice bitov, ktoré môžu byť počas procesu modifikované. Ak nebola vložená správa podobná (v sledovanej charakteristike) nosiču, je možné pozorovať markantné rozdiely vypočítaných charakteristík. Pokiaľ by sa na výber miest použil náhodný prístup, nezávisle od podobnosti správy s nosičom by boli charakteristiky podobné.

vzhľadom na obsah nosiča: Výber miest budúcej zmeny závisí od obsahu nosiča vzhľadom na vkladanú správu. Ak by sme uvažovali predošlý príklad modifikácie najmenej významného bitu, tento typ výberu by mohol byť realizovaný nasledovným spôsobom. Ak chceme zistiť, či modifikovať najmenej významný bit skúmaného obrazového bodu, do rozhodovacieho procesu zahrnieme napríklad okolité body. Určíme, nakoľko sú okolité farebné body rôznorodé (nakoľko sa farebne odlišujú). Ak táto odlišnosť presiahne vopred definovanú mieru, skúmaný obrazový bod možno použiť na vloženie jedného bitu informácie, v opačnom prípade sa bude skúmať ďalší kandidát. Takýmto spôsobom je možné zabrániť narušeniu uvažovaného steganografického systému vizuálnou kontrolou, nakoľko sa algoritmus vyhýba modifikácií obrazových bodov, ktoré tvoria farebné monotónne plochy.

[7] jestvujú však steganografické systémy, v ktorých tento typ výberu eliminuje výrazným spôsobom detekciu správy

Bezpečnosť steganografickej schémy

Nakoľko je v literatúre najrozšírenejšou definíciou steganografickej bezpečnosti definícia od Cachina [10], prevzali sme ju aj my pre účely tejto práce. Vychádza zo základov teórie informácie. Nie je však jediná. Jestvujú mnohé ďalšie prístupy, ktorými je možné definovať steganografickú schému a jej bezpečnosť. Pre zmienku spomeňme napríklad prístupy, ktoré skúmajú bezpečnosť z pohľadu vlastností detektora [11], alebo vychádzajú z pravdepodobnostnej teórie hier [45], alebo skúmajú model steganografickej schémy so zašumeným kanálom [36].

Nech P_C je dané rozdelenie pravdepodobnosti a nech $c \in \mathscr{C}$. Potom výraz $P_C(c)$ určuje pravdepodobnosť výberu nosiča c podľa daného rozdelenia P_C.

Význam pojmu rozdelenia pravdepodobnosti v steganografii ukážeme na tradičnom príklade tajnej komunikácie väzňov. Predpokladajme, že dozorca dovolil posielať vzájomne medzi väzňami ich rodinné fotografie. Nech rozdelenie pravdepodobnosti P_C je dané vedomosťami dozorcu, ktoré fotografie patria medzi rodinné. Potom ak jeden z väzňov pošle fotografiu, na ktorej sa nachádzajú nejakí členovia jeho rodiny, u dozorcu fotografia nevzbudí žiadne podozrenie, pretože pravdepodobnosť, že táto fotografia je rodinnou fotografiou, sa vyhodnotí ako nenulová. Pokiaľ však niekto z väzňov pošle fotografiu Michelangelovej sochy Dávida, táto sa bude zdať dozorcovi veľmi podozrivá, nakoľko je príliš pochybné, aby bol odosielateľ v pokrvnom príbuzenstve s Dávidom.

V mnohých prípadoch je určenie rozdelenia pravdepodobnosti pre nosiče, ktoré neobsahujú steganografickú informáciu, príliš zložité. Vtedy je možné použiť negatívny prístup a definovať rozdelenie pre steganografické médiá. V mnohých prípadoch je oveľa jednoduchšie vyjadriť, aké vlastnosti má médium obsahujúce steganografickú informáciu, než vyjadriť vlastnosti čistého nosiča. Napríklad pri steganografických systémoch statických obrázkov nedokážeme vyjadriť rozdelenie pravdepodobnosti „normálnosti“ všetkých typov obrázkov, pretože nevieme definovať tie vlastnosti obrazu, ktoré konštituujú jeho „normálnosť“ (v zmysle, že sa v obrázku nenachádza steganografická správa). Pre konkrétne funkcie vkladania správy však vieme definovať vlastnosti, ktoré odlišujú steganografické médium od čistého. Napríklad v prípade steganografickej techniky modifikácie najmenej významného bitu je možné odlíšiť steganografický nosič od čistého na základe histogramu tzv. blízkych párov.

Nech pre nami definovanú steganografickú schému platí, že:

1. výber nosiča je realizovaný podľa rozdelenia pravdepodobnosti P_C,
2. výber kľúča k a správy m je realizovaný náhodne s rovnomerným rozdelením,

a P_S označuje náhodné rozdelenie steganografických nosičov. Potom definujeme nasledovné hypotézy:

1. H_0: nosič $c \in \mathscr{C}$ neobsahuje steganografickú správu,
2. H_1: nosič $c \in \mathscr{C}$ obsahuje steganografickú správu.

Ak je pravdivá hypotéza H_0, potom realizácia výberu média $c \in \mathscr{C}$ je daná rozdelením P_C; ak je pravdivá hypotéza H_1, potom realizácia výberu média $c \in \mathscr{C}$ je daná rozdelením P_S. Pri takto definovaných hypotézach **chyba prvého druhu** nastáva vtedy, ak sa testom zamietne hypotéza H_0, hoci platí. To znamená situáciu, v ktorej vyhlásime médium za

steganografické, hoci je čisté. Maximálna prípustná veľkosť chyby prvého druhu sa nazýva **hladinou významnosti testu** a túto pravdepodobnosť označujeme symbolom α. **Chybou druhého druhu** rozumieme pravdepodobnosť, s akou hypotéza H_0 neplatí, ale testom ju nezamietneme; teda rozhodneme, že testované médium neobsahuje steganografickú správu, hoci ju obsahuje. Vzniká v dôsledku neschopnosti testu zistiť, že nulová hypotéza neplatí. Túto pravdepodobnosť označujeme symbolom β. Doplnok chyby druhého druhu do jednotky $(1-\beta)$ nazývame **sila testu**. Sila testu určuje pravdepodobnosť, s akou hypotézu H_0 zamietneme, keď neplatí. Teda vyjadruje pravdepodobnosť, s akou sme schopní odhaliť neplatnosť hypotézy.

Porovnanie dvoch náhodných rozdelení môžeme realizovať pomocou vzťahu, ktorý vyjadruje **relatívnu entropiu** porovnávaných rozdelení pravdepodobnosti:

$$\mathrm{D}(P_C||P_S) = \sum_{c\in\mathscr{C}} \left(P_C(c) * \log_2 \frac{P_C(c)}{P_S(c)} \right) \tag{1.1}$$

Ak máme náhodné premenné P_C a P_S s alternatívnym rozdelením ($P_C = (p, 1-p)$ a $P_S = (q, 1-q)$), ich vzájomnú relatívnu entropiu vyjadríme pomocou vzťahu 1.1 nasledovne:

$$\mathrm{D}(P_C||P_S) = p * \log_2 \frac{p}{q} + (1-p) * \log_2 \frac{1-p}{1-q} \tag{1.2}$$

Hovoríme, že steganografická schéma je *dokázateľne bezpečná*, ak $\mathrm{D}(P_C||P_S) = 0$. Ak ale platí $\mathrm{D}(P_C||P_S) \leq \epsilon$, steganografická schéma je ϵ–bezpečná.

Výraz $\mathrm{D}(P_C||P_S)$ je rovný nule vtedy a len vtedy, ak platí, že obe porovnávané rozdelenia sú rovnaké. V našom prípade to znamená situáciu, keď $p = q$. Vtedy je náhodné rozdelenie čistého nosiča zhodné s náhodným rozdelením steganografického média, a pozorovateľ nie je schopný odlíšiť, či ide o nosič s vloženou správou alebo nie.

Ďalej definujme *binárny detektor*, ktorého výstupom je hodnota 1 v prípade, že pozitívne rozhodne o steganografickom obsahu média. V opačnom prípade (keď ide o čisté médium), je jeho výstupom 0. Rozhodovanie tohto binárneho detektora pre alternatívne rozdelenia P_C a P_S je potom dané nasledovne:

- Ak predpokladáme na vstupe detektora iba čisté médiá (s náhodným rozdelením P_C), s pravdepodobnosťou $1-\alpha$ detektor rozhodne, že je médium čisté, a s pravdepodobnosťou α rozhodne nesprávne.
- Ak na vstupe detektora predpokladáme iba steganografické médiá s náhodným rozdelením P_S, s pravdepodobnosťou $1-\beta$ detektor rozhodne správne, že médium obsahuje skrytú správu. S pravdepodobnosťou β rozhodne nesprávne.

Potom je vzájomná *binárna relatívna entropia* chýb prvého a druhého druhu detektora modelovaná náhodnými premennými s alternatívnym rozdelením $(\alpha, 1-\alpha)$ a $(\beta, 1-\beta)$ daná (pomocou vzťahu 1.2) ako funkcia dvoch premenných nasledovne:

$$r(\alpha,\beta) = \alpha * \log_2 \frac{\alpha}{1-\beta} + (1-\alpha) * \log_2 \frac{1-\alpha}{\beta} \tag{1.3}$$

Ak zvolíme $\alpha = 0$, potom z definície ϵ–bezpečnej steganografickej schémy vyplýva:

$$\log_2 \frac{1}{\beta} \leq \epsilon \tag{1.4}$$

V takto zvolenej steganografickej schéme potom vieme ohraničiť chybu druhého druhu:

$$\beta \geq 2^{-\epsilon}$$

To znamená, že čím menšie ϵ zvolíme, alebo čím podobnejšie sú náhodné rozdelenia P_C a P_S, tým je väčšia pravdepodobnosť, že správa nebude detektorom objavená.

Najjednoduchším systémom, ktorá spĺňa vyššie uvedenú Cachinovu definíciu dokázateľnej bezpečnosti steganografickej schémy, je obdoba jedinej dokázateľne bezpečnej šifry. Jedná sa o systém, ktorého kódovacia funkcia tvorí údajovú reprezentáciu nosiča. Nie je problém funkciu vkladania upraviť tak, aby len modifikovala štruktúru nosiča. Nech $\mathscr{C} = \{0,1\}^n$ a P_C je rovnomerné rozdelenie na množine $\mathscr{C}$. Nech $\mathscr{K} = \{0,1\}^n$, $k \in \mathscr{K}$ je jednorazová náhodná postupnosť a nech $\mathscr{M} = \{0,1\}^n$, $m \in \mathscr{M}$. Potom funkcia vkladania správy do média je $s = k \oplus m$ a na extrakciu správy sa použije funkcia $m = k \oplus s$.

Značnou nevýhodou tohto systému je potreba bezpečného prenosu jednorazového kľúča rovnakej dĺžky, ako má správa. Okrem malej efektívnosti tohto systému je veľmi vážne poškodená požiadavka nenápadnosti média. Nedostatkom Cachinovej definície bezpečnosti steganografickej schémy je to, že nezohľadňuje mieru podozrivosti samotného nosiča voči prostrediu prenosového kanála (keďže prenosový kanál nie je v nami uvažovanej steganografickej schéme definovaný).

1.3 Najrozšírenejšie pojmy steganografie

Textovo orientovaná steganografia

Do tejto skupiny zaraďujeme všetky techniky ukrývania informácie, ktoré sú závislé od textu. Pritom je jedno, či sa jedná o techniky založené na vonkajších charakteristikách textu (veľkosti interpunkčných znamienok, medzery medzi slovami alebo riadkami), alebo o techniky, ktoré sledujú vnútorné závislosti jazyka, v ktorom je text napísaný (syntaktické a sémantické metódy steganografie). Techniky, ktoré sledujú vonkajšie znaky tlačeného textu, je možné realizovať vďaka veľkej presnosti súčasnej tlače (rozlíšenie jednotlivých tlačených bodov od 300 do 1600 dpi a viac). K procesu dekódovania vloženej informácie pomocou techník meniacich formátovanie dokumentu je potrebný buď pôvodný dokument (bez vykonaných zmien formátovania), alebo opis formátovania nemodifikovaného dokumentu.

Techniky medzier medzi riadkami a slovami: Uvedená technika je chránená patentom US 5629770 z 13. mája 1997, s platnosťou do 31. marca 2015[8]. Hlavná myšlienka spočíva v kódovaní informácie pomocou vzájomných posunov slov a/alebo riadkov. Podobne možno sledovať posun interpunkčných znamienok vpravo/vľavo od ich zvyčajného umiestnenia.

Techniky malých zmien vlastností textu: Každé písmeno textu má dostatočné množstvo špecifických čŕt, ktoré je možné použiť na kódovanie tajnej správy. Napríklad: šírka a výška písmena, pomer šírky a výšky, polomery konkáv a girland niektorých písmen...

[8] http://www.patentstorm.us/patents/5629770/claims.html, dostupné 11. augusta 2011

Steganografia vo zvukovom zázname

Vkladanie tajnej správy do zvykového záznamu je pre steganografiu veľkou výzvou. Ľudský sluchový systém je značne dynamický — dokáže rozlišovať silu zvuku v rozsahu približne milióna jednotiek a vnímať najvyššiu frekvenciu, ktorá je rádovo tisíc násobkom najnižšej. Preto nie je jednoduché vložiť do zvukového záznamu informáciu tak, aby človek, ktorý tento záznam bude počúvať, neodlíšil médiá obsahujúce tajnú informáciu od nemodifikovaného zvukového záznamu. Jedinou doteraz známou možnosťou, ako využiť zvuk na prenos tajnej informácie, spočíva vo využití zvýraznenia zvolených častí pôvodnej zvukovej informácie.

Pred samotným kódovaním tajnej informácie je potrebné zvážiť dva činitele, ktoré ovplyvňujú proces vkladania:

1. digitálny formát zvukového záznamu a
2. médium, do ktorého bude záznam uložený.

Zvyčajne sa na kódovanie zvukového záznamu používajú nasledovné techniky:

Lineárna kvantizácia: Jednoduchá lineárna 16 bitová vzorkovacia metóda využívaná napríklad v súboroch typu WAV.

Dočasná vzorkovacia frekvencia: Používa vybrané pásma frekvencie ako body vzorkovania zvukovej informácie.

Vnemové vzorkovanie: Táto technika na základe štatistických metód modifikuje vstupný signál tak, že z neho vynecháva časti, ktoré počúvajúci nie je schopný vnímať. Používa sa pre najlepší pomer informácie k objemu uchovávaných údajov a nachádza sa v základných štandardoch, ktoré špecifikujú procesy stratovej kompresie multimediálnych signálov.

Prenosový kanál zvukovej informácie (ktorá sa šíri od vysielača k prijímaču) je možné rozdeliť na nasledovné typy (spracované podľa [6]):

1. digitálny prenos bezo zmien prenášanej informácie,
2. prevzorkovanie digitálneho signálu,
3. zmena digitálneho signálu na analógový a jeho prevzorkovanie,
4. rádiový prenos zvuku do koncového zariadenia (reproduktoru) a jeho následné vzorkovanie pomocou vstupného zariadenia (mikrofónu).

V nasledujúcom texte spomenieme tri najznámejšie metódy vkladania informácie do zvukového signálu: zmenu najmenej významného bitu, kódovanie pomocou fázy a spektrálne kódovanie.

Kódovanie pomocou zmeny najmenej významného bitu je síce najjednoduchšie na implementáciu, ale nie je dostatočne odolné voči odhaleniu. Aj napriek zmene iba najmenej významnej zložky zvukovej informácie je táto zmena pozorovateľná (vďaka na začiatku tejto časti spomínanej schopnosti ľudského zvukového systému rozlíšiť veľké množstvo zmien intenzity a frekvencie).

Kódovanie pomocou fázy nahrádza počiatočnú fázu zvukovej informácie takou, ktorá reprezentuje tajnú informáciu. Táto technika je silne previazaná na diskrétnu kosínusovú transformáciu, nakoľko realizácia uvažovanej techniky sa vykonáva na koeficientoch tejto transformácie.

Spektrálne kódovanie vkladá informáciu do celého frekvenčného rozsahu zvukového signálu. Samotný zvuk je prenášaný na rozličných frekvenciách v závislosti od použitej metódy spektrálneho kódovania. Jednou z používaných metód je napríklad DSSS (*Direct Sequence Spread Spectrum*) . Táto metóda rozkladá nosný signál do spektra jeho násobením s pseudonáhodnou postupnosťou. Metódy spektrálneho kódovania sú zo spomenutých najbezpečnejšie vzhľadom na detekciu ukrytej správy, ale zavádzajú do signálu šum, čím zvyšujú možnosť straty údajov.

Steganografia vo video súboroch

Techniky ukrývania tajnej informácie vo video súboroch využívajú steganografické techniky statického obrazu (informácie o nich sa nachádzajú v kapitole 3 tohto dokumentu na strane 28). Rozdiel je v kapacite nosiča. Do pohyblivého obrazu je možné vložiť rádovo viac informácie ako do statického obrazu. Je to dané technikou tvorby videa; pohyblivý obraz je sledom statických obrazov. Aj napriek mnohým technikám kódovania pohybu (sledovanie iba relatívnych zmien, vektorový popis objektov obrazu...) je možné využívať techniky vkladania tajnej informácie pomocou modifikovaných metód steganografie v statických obrazoch.

Doteraz jediný verejne dostupný softvérový nástroj (s názvom `StegoVideo`), ktorý implementuje steganografiu do pohyblivého obrazu, vyvinula spoločnosť MSU[9].

Steganografia v statických obrazoch

V súčasnosti sú techniky vkladania tajnej informácie do statických obrázkov najrozšírenejšími a najskúmanejšími. Je to vďaka nasledovným skutočnostiam:

- obrázky obsahujú veľké množstvo redundantnej informácie (a často aj šumu),
- schopnosť ľudského zrakového ústrojenstva rozlíšiť malé zmeny v obraze je nízka (oveľa nižšia ako napríklad pri zvukovom — viď časť o steganografii vo zvukovom zázname na strane 16).

Viac o tomto type steganografie sa nachádza v kapitole 2 (str. 21) tohto dokumentu.

Iné formy steganografie

Okrem spomenutých foriem steganografie jestvuje mnoho ďalších. Nie je možné vymenovať všetky, nakoľko ľudská vynaliezavosť je schopná využiť akýkoľvek komunikačný kanál, ktorý obsahuje redundantnú alebo nevyužitú informačnú kapacitu, na prenos utajenej informácie. Pre ilustráciu tejto fantázie a vynaliezavosti predkladáme pár ukážok z oblasti využitia operačných systémov a sieťovej komunikácie.

[9] aplikácia a jej popis je dostupný na http://compression.ru/video/stego_video/index_en.html, dostupné 11. augusta 2011

Steganografia v súborovom systéme

Počiatky steganografie, ktorá využíva štruktúry operačného systému, ktoré slúžia na správu rozloženia údajov na prepisovateľnom médiu (zvyčajne sa jedná o pevný disk), siahajú k systému súborov FAT (*File Allocation Table*). Tento systém tvorí jednorozmerná tabuľka (podľa nej má systém názov), ktorá uchováva informácie o obsadenosti jednotlivých častí disku. Disk je logicky rozčlenený na malé úseky (sektory) a FAT uchováva informáciu o použití týchto sektorov.

Steganografia založená na systéme FAT ukladá tajnú informáciu do nevyužitých oblastí (sektorov) disku. Nakoľko sa jedná o sektory označené operačným systémom ako voľné, žiadne systémové ani používateľské údaje sa nezničia. Problémom je situácia, keď používateľ zapíše na disk nové údaje. Keďže operačný systém, ktorý prideľuje miesto údajom na disku nevie nič o uloženej tajnej informácii, môže sa stať, že prepíše tajnú informáciu. S týmto systémom sa rôzne steganografické súborové systémy vyrovnávajú odlišne. Najjednoduchším riešením je náhodná voľba voľných miest na disku, kam sa tajná informácia uloží a duplicita uloženej informácie. Predpokladom úspešnosti takého systému je v prvom rade dostatok voľného miesta na disku a také správanie operačného systému, ktoré sa snaží prideľovať novým zapisovaným údajom na disku súvislé miesto. Tento predpoklad je pri súborovom systéme FAT splnený, nakoľko uvedené správanie vedie k zvýšeniu priepustnosti systému pri potrebe načítať súbor (hlava disku sa nemusí presúvať medzi viacerými oblasťami disku, ale môže načítať celý súbor v jednom neprerušenom slede operácií). Potom v prípade, že je tajná informácia roztrúsená náhodne po celom disku, sa pri prípadnom prepísaní nejakého sektoru údajmi operačného systému nezničí celá tajná informácia, ale iba jej malá časť. Potom je možné obnoviť informáciu zo záložného sektora, nakoľko tajná informácia je na disku uložená duplicitne.

Steganografia v sieťových protokoloch

Rodina protokolov zásobníka TCP/IP v sebe zahŕňa veľa možností, ako implementovať steganografiu. Na tieto účely sa využívajú predovšetkým rôzne údajové štruktúry (napríklad príznakové a vyhradené polia) v hlavičkách jednotlivých protokolov. V nasledujúcom texte spomenieme iba najrozšírenejšie protokoly (IP verzia 4 a TCP) a ich steganografické možnosti využitia.

V protokole IPv4 je možné využiť niekoľko bitov poľa TTL (*Time To Live*). Toto pole určuje dĺžku prežitia datagramu počas transportu sieťou. Hodnota poľa sa znižuje (o jedna) každým prechodom nejakého smerovača na ceste. Veľkosť poľa TTL je 8 bitov. Pokiaľ vieme, že počet smerovačov počas prenosu datagramu nepresiahne hodnotu 2^M, na prenos tajnej informácie môžeme v jednom datagrame využiť v rámci poľa TTL $8 - M$ bitov.

Ďalej je možné v IPv4 hlavičke využiť informatívne pole protokolu vyššej vrstvy (8 bitov). Pomocou tohto poľa môžeme prenášať údaje iba v tom prípade, že na strane prijímateľa je možné odchytiť datagram ešte pred jeho spracovaním operačným systémom. Operačný systém totiž zahadzuje tie datagramy, pre ktoré cieľový protokol vyššej vrstvy nie je podporovaný. Odchytávanie datagramov umožňuje napríklad knižnica *libpcap*.

V prípade, že sa na strane príjemcu nefiltruje IP adresa (identifikátor v celosvetovej pavučine) odosielateľa, je možné použiť na prenos tajnej informácie jej údajovú štruktúru v záhlaví IPv4 protokolu. Na adresu (prijímateľa i odosielateľa) sú vyhradené štyri bajty (32 bitov), takže je možné jedným prenosom získať 32 bitov tajnej informácie.

40 bajtov dlhé pole s voliteľnými položkami protokolu TCP býva veľmi zriedka filtrované (bežne ho využívajú aplikácie na udržanie TCP spojenia), preto je vhodné na prenos údajov. V privátnych sieťach, kde sa kladie veľký dôraz na bezpečnosť, však môže byť toto pole kontrolované.

Zdrojový a cieľový port komunikácie protokolu TCP je možné využiť na prenos tajnej informácie iba v prípade, že aplikácia dokáže odchytiť datagram skôr, než ho spracuje operačný systém. Podobne je možné využiť bajt pre poradové číslo odosielaného (alebo prijímaného) datagramu. Ale to iba v prípade, že nevyužívame spojovo orientovaný prenos údajov, pretože inak by sa zmenou poradového čísla rušil spojový prenos.

Pomocou ukazovateľa naliehavých údajov je možné prenášať dva bajty informácie. Ukazovateľ sa využíva iba v prípade nastavenia príznaku `URG`, preto ho treba pred prenosom nastaviť.

1.4 Steganoanalýza

Steganoanalýza je obdobou kryptoanalýzy — jej primárnym cieľom však nie je získať tajnú správu z komunikačného kanála, ale potvrdiť (alebo zamietnuť) jej prítomnosť v skúmanom nosiči. To, že nie je potrebné extrahovať správu z nosiča, je prirodzeným dôsledkom záujmu steganografie. Steganografia sa snaží ukryť existenciu tajnej komunikácie, nejde jej o to urobiť správu nečitateľnou nelegitímnym účastníkom komunikácie (to je cieľom kryptografie).

V časti Prenosový kanál na strane 10 sme definovali tri typy nelegitímneho účastníka komunikácie v steganografickom systéme. Podľa toho, aké informácie má pozorovateľ (potenciálny narušiteľ) systému o prebiehajúcej tajnej komunikácie, rozdeľujeme steganoanalýzu na dve základné skupiny: cielenú a necielenú[10].

- O *cielenej steganoanalýze* hovoríme vtedy, ak má narušiteľ steganografického systému k dispozícií informácie o použitej steganografickej schéme a jej implementácii.
- V prípade, že útočník má iba isté podozrenie ohľadom realizácie skrytého prenosu, musí vymyslieť dostatočný počet steganoanalytických algoritmov, aby bol schopný detekovať akúkoľvek skrytú komunikáciu. Nejde o steganografickú analýzu zameranú na nejaký konkrétny steganografický systém, preto tento typ analýzy nazývame *slepá steganoanalýza*.

Vyššie uvedené skupiny je možné rozčleniť na jednotlivé typy a urobiť nasledovné rozdelenie steganoanalytických útokov, pri ktorých je k dispozícii:

1. iba nosič, obsahujúci steganografickú informáciu (*stego only attack*),
2. steganografické aj čisté médium (*known cover attack*),
3. správa a médium obsahujúce túto správu (*known message attack*),
4. zvolená správa a nosič obsahujúci túto správu (*chosen message attack*),
5. použitý steganografický algoritmus a steganografické médium (*chosen stego attack*),
6. steganografický algoritmus, ľubovoľná správa a k nej prislúchajúce steganografické médium (*known stego attack*).

[10] v literatúre sa necielená steganoanalýza zvyčajne označuje pojmom „slepá“

Steganoanalýza patrí medzi klasifikačné problémy. Pre dané médium sa určuje príslušnosť buď do množiny nosičov, ktoré tajnú informáciu obsahujú, alebo neobsahujú. Preto sa v procese analýzy využívajú dostupné prostriedky strojového učenia, rozlišovania vzoriek a klasifikácie. Stále viac sa v tejto oblasti využívajú formalizmy neurónových sietí, umelej inteligencie, genetických a evolučných výpočtov, Petriho sietí...

Pokiaľ model klasifikácie obsahuje iba dve disjunktné množiny, používame miesto pojmu klasifikácia výraz detekcia. Zariadenie, ktoré proces detekcie realizuje, nazývame detektor. Úspešnosť detekcie je podmienená voľbou množín, rozhodovacím algoritmom detektora a chybami, ktorých sa detektor dopúšťa (falošný poplach — keď klasifikuje čisté médium ako steganografické; a chybná detekcia — keď steganografické médium zatriedi do množiny čistých).

V časti 1.2 sme definovali steganografický systém a pre určenie bezpečnosti steganografickej schémy sme vychádzali z definície Cachina. Bezpečnosť steganografického systému veľmi úzko súvisí s oblasťou steganoanalýzy: steganografická analýza preveruje bezpečnosť systému. V našej práci je bezpečnosť uvedená vzhľadom na vlastnosti náhodných rozdelení výberu nosiča a steganografického média. Na takto definovanú bezpečnosť však nie je možné skonštruovať ideálny detektor. Nepoznáme náhodné rozdelenie výberu nosiča. Preto sa pri tvorbe detektora postupuje tak, že sa skonštruuje zobrazenie z množiny nosičov do množiny charakteristík, ktoré sa využívajú v procese klasifikácie. Vytvorí sa množina vlastností, ktoré reprezentujú nosiče. Táto voľba je najcitlivejším miestom pri návrhu detektora. Ak sa vyberú vhodné vlastnosti nosičov, ktoré ich budú v procese klasifikácie reprezentovať, zvýši sa správna rozlišovacia schopnosť detektora, a zníži sa počet chýb falošného poplachu a chybnej detekcie.

Na zobrazenie, ktoré mapuje priestor nosičov do nejakého priestoru charakteristík je kladená požiadavka, aby v čo najlepšej miere mapovala každé médium na jemu zodpovedajúci bod v priestore charakteristík. Najlepšou situáciou je, ak je toto zobrazenie bijektívne. Vtedy je možné klasifikovať každý bod priestoru charakteristík do jednej z dvoch množín (jedna množina pre čisté, druhá pre steganografické nosiče). Ak by bolo zobrazenie iba prosté (niektorým bodom priestoru charakteristík by nezodpovedalo žiadne médium), návrh aj funkcia generátora by mohli ostať rovnaké ako v prípade bijektívneho zobrazenia. Obmedzenie na bijektívne zobrazenie iba zjednodušuje teoretickú analýzu detektora. Keď ide o prosté zobrazenie, nijako sa nemení funkčnosť detektora — ten mapuje nosiče do priestoru charakteristík, a v ňom je každému médiu priradený práve jediný bod. Problematické sú surjektívne zobrazenia, pretože ak sa mapuje zároveň čisté aj steganografické médium na ten istý bod priestoru charakteristík, zavádza sa návrhová (systémová) chyba falošného alarmu a/alebo chybnej detekcie. Ak sa takému návrhu nedá vyhnúť, cieľom je minimalizovať množinu kolíznych bodov priestoru charakteristík (čím sa zároveň minimalizujú oba typy chýb detektora).

Kapitola 2

Steganografia v statickom obraze

Ako bolo spomínané v stati Steganografia v statických obrazoch na strane 17, steganografia v statických obrázkoch patrí k najpreskúmanejším oblastiam v oblasti prenosu tajnej informácie na pozadí bežnej komunikácie.

Prv, než objasníme techniky kódovania tajnej správy v obrázkoch, v skratke vysvetlíme štruktúru a kompresiu najpoužívanejších grafických súborových formátov. Potom sa budeme zaoberať najrozšírenejšími steganografickými technikami.

2.1 Štruktúra grafických súborových formátov

„Počítaču sa obrázok javí ako pole čísel, ktoré reprezentuje intenzitu svetla jednotlivých bodov obrazu.“ (Duncan Sellars)

V steganografii sa bežne pracuje s 8 alebo 24 bitovými obrázkami. Pri 8 bitových obrázkoch je jeden obrazový bod popísaný ôsmymi údajovými bitmi, pri 24 bitových dvadsiatimištyrmi bitmi. Výhody a nevýhody oboch typov budú objasnené v nasledujúcom texte.

8-bitový obrazový formát sa používa kvôli malej výslednej veľkosti použitého úložného priestoru (súboru), ktoré popisuje grafický obrázok. Nevýhodou je malý počet farieb (iba 256), ktoré sú dostupné na popis farebného statického obrazu. Preto sa tento formát v súčasnosti používa takmer výhradne na ukladanie obrazu v šedej škále. Používanie šedo–tónovaného obrazu má v steganografii výhodu oproti farebnému v tom, že je ťažšie detekovať malú zmenu farby obrazového bodu v šedej škále ako vo farebnej.

24-bitový farebný model dokáže kódovať 2^{24} (približne 16 miliónov) farieb. Výhodou tohto modelu oproti 8-bitovému je skutočnosť, že pojme oveľa väčšie množstvo ukrývaných údajov. Na druhej strane nevýhodou je veľkosť výsledných súborov: pri 24 informačných bitoch sú to rádovo megabajty, pri 8 bitoch kilobajty.

2.2 Kompresia obrazových údajov

Kompresiou sa chápe taký proces kódovania informácie, ktorého výstupom je menší objem informácie, než ten pôvodný. V oblasti digitálneho spracovania obrazu sa používajú dva základné typy kompresie:

1. stratová a
2. bezstratová.

Stratová kompresia vo veľkej miere redukuje pôvodný objem grafickej informácie, avšak na úkor kvality obrazu. Pri algoritmoch tohto typu sa využíva nedokonalosť ľudského vizuálneho systému, ktorá umožňuje vynechať z obrazu tie informácie, ktoré ľudské oko nie je schopné spracovať (napríklad množstvo odtieňov alebo prílišné detaily). Kompresia je stratová, čo znamená, že po aplikácii algoritmu kompresie nie je možné obnoviť pôvodnú grafickú informáciu, nakoľko nadbytočné údaje (to znamená tie, ktoré sú ľudským vizuálnym systémom nepostrehnuteľné) boli z originálu odstránené.

Bezstratová kompresia, ako už samotný názov hovorí, neodstraňuje nič z pôvodnej grafickej informácie, ale snaží sa využiť algoritmy jej optimálneho kódovania. Značnou nevýhodou je rôznorodosť obrazovej informácie, čo spôsobuje iba nízke znižovanie pôvodného objemu údajov.

Podľa výskumov uvedených v [17] je ľudské oko oveľa citlivejšie na malú zmenu jasu než farby. Preto na reprezentáciu obrazu nie je potrebné uchovávať informáciu o troch farbách (ktorých zmiešaním je daná farebná hodnota obrazového bodu), ale stačí informácia o luminiscenčnej a chromatickej zložke farebného spektra. Keďže je ľudské oko viac citlivé na zmenu jasu (luminiscencia), chromatickú zložku netreba uchovávať s takou presnosťou ako zložku jasu. Túto vlastnosť vnímania ľudského vizuálneho systému využíva kompresia `JPEG` na prvotné zníženie objemu údajov, ktoré sa budú spracovávať (minimálne na polovicu). Dosahuje sa to zmenou reprezentácie z farebného modelu `RGB` na `YUV`.

V druhom kroku sa pomocou diskrétnej kosínusovej (alebo aj Fourierovej) transformácie (skratkami DCT alebo DFT) zmení reprezentácia obrazu z priestorovej do frekvenčnej oblasti. Operácia sa vykonáva na blokoch o štandardnej veľkosti 8x8 obrazových bodov. Výstupom je 64 koeficientov. Zmena ktoréhokoľvek koeficientu sa prejaví na všetkých 64 obrazových bodoch.

Tretím krokom kompresie je kvantizácia. Ľudské oko je síce najcitlivejšie na jas obrazu, ale nie rovnomerne. V oblasti vysokých frekvencií sú potrebné väčšie zmeny, aby ich ľudský zrakový systém vnímal. Táto vlastnosť nelinearity sa používa na elimináciu vysokých frekvencií spektra, ktoré človek nedokáže vnímať (alebo ktorých neprítomnosť neznižuje vizuálnu kvalitu obrázka). Eliminácia koeficientov sa robí násobením matice skalárom (kvantizačným koeficientom).

Nakoniec sa hodnota prvkov výslednej matice zakóduje pomocou Huffmanovho kódovania. Keď sa v treťom kroku algoritmu urobí kvantizačná úprava, mnohé koeficienty nadobudnú nulovú hodnotu. Potom sa kódujú iba nenulové prvky matice (od ľavého horného rohu — od najnižších frekvencií). Zmysel `JPEG` kompresie spočíva v redukcii údajov pomocou kvantizácie.

2.3 Steganografické techniky v statickom obraze

Využitie najmenej významného bitu

Algoritmy modifikujúce najmenej významný bit patria do triedy techník, ktoré robia zmeny v priestorovej oblasti obrazu.

Pri ukrývaní informácie do obrazu sa využíva nedokonalosť ľudského optického rozlišovania malých zmien farebných tónov. Dve farby sa v bitovej reprezentácií najmenej líšia pri zmene najmenej významného bitu. Tento sa označuje LSB z anglického *Least Significant Bit*. Technika ukrývania informácií do obrazovej mapy zmenou najmenej významného bitu sa označuje ako LSB technika.

Jednotlivo zoskupené farebné zložky obrazových bodov tvoria tzv. *farebné roviny*. Vzhľadom na skutočnosť, že farba obrazového bodu je určená niekoľkými zložkami, jestvuje viacero variácií algoritmu: je možné sekvenčné vkladanie správy postupne do každej farebnej roviny zvlášť, alebo cyklické striedanie rovín v istom poradí. Takisto je možné modifikovať algoritmus na náhodné vkladanie: pri tomto postupe sa nevkladajú bity správy do každého LSB bitu postupne za sebou, ale podľa zvolenej náhodnej postupnosti sa isté LSB bity ponechávajú bezo zmeny a správa sa vkladá iba do farebných bodov určených vygenerovanou náhodnou postupnosťou.

Do jedného obrazového bodu je možné uložiť v základnej verzii algoritmu tri bity správy. V súlade s definíciou 2 na strane 9 vieme určiť kapacitu média vopred podľa rozmerov obrázka, do ktorého sa správa vkladá. Označme šírku obrazu symbolom w a výšku h. Oba rozmery sa udávajú v obrazových bodoch. Potom kapacita média cap je daná vzťahom:

$$cap = w * h * 3$$

Pre vyjadrenie v bajtoch je potrebné výslednú hodnotu deliť ôsmymi (keďže podľa definície je kapacita udávaná v bitoch).

Technika LSB pre obrazy určené paletou farieb

Nie všetky typy reprezentácií obrazu používajú v priestorovej doméne mapu obrazových bodov, v ktorej má každý bod priamo definované hodnoty jednotlivých farebných zložiek. Niektoré reprezentácie používajú na mieste popisu farieb odkazy do tabuľky farieb. Výhodou tohto prístupu je menšia veľkosť reprezentácie obrazu. Stačí, ak sa uložia použité farby v obraze do jednorozmerného poľa, a v mape obrazových bodov sú hodnoty farieb uložené ako indexy do tabuľky použitých farieb. Technika indexovania je vhodná pre obrazy, v ktorých sa nevyužívajú v mape obrazových bodov tri farebné zložky (z ktorých každá používa na reprezentáciu 8 bitov). Počtu bitov, ktoré sú použité na reprezentáciu farby jedného obrazového bodu sa hovorí *farebná hĺbka obrazu*.

Obrazový formát `GIF` môže použiť maximálne 8-bitovú farebnú hĺbku. Ide o obmedzenie samotnej reprezentácie grafickej informácie. To znamená, že obrázok vo formáte `GIF` môže obsahovať nanajvýš 256 farieb [20]. Farby uložené v palete sú zvyčajne usporiadané podľa frekvenčného výskytu od najviac používaných po najmenej. Je to kvôli tomu, aby sa minimalizoval čas prístupu k jednotlivým položkám tabuľky pri spracovaní obrazových údajov [44]. Usporiadanie farieb v palete podľa frekvencie ich použitia sťažuje použitie LSB algoritmu. Nie je možné jednoducho meniť najmenej významný bit indexu, pretože takáto zmena môže mať za následok viditeľnú zmenu farby obrazového bodu. Dve susedné farby použitej palety nemusia mať hodnotu rozdielu menšiu než je istá hranica vnímateľnosti ľudského vizuálneho systému.

Je možné použiť techniku LSB aj v prípade indexovaných obrazových súborov. Modifikácia spočíva v poprehadzovaní farieb v palete tak, aby boli usporiadané podľa najmenších rozdielov farebných hodnôt príslušných dvojíc indexov, ktoré sa algoritmom LSB na seba vzájomne mapujú (ide vždy o dvojicu farieb v palete na párnom a nasledujúcom nepárnom mieste). Táto technika nového usporiadania farieb v palete môže byť dostatočne účinná, ale je ľahko detegovateľná. Detektor na prítomnosť steganografickej informácie môže iba jednoducho overiť, či sú farby palety usporiadané podľa frekvenčného výskytu. Ak nie sú, rozhodne, že médium obsahuje steganografickú správu. Správu je možné zničiť preusporiadaním palety podľa frekvencie použitých farieb (s tým súvisí zmena indexov obrazovej informácie).

Steganografia JPEG využívajúca štruktúru nosiča

Podobne ako pri ostatných binárnych súborových formátoch, aj súbor `JPEG` je definovaný pomocou nejakej štruktúry. Na ukladanie údajov sa používa vkladanie skupín bajtov, tzv. **značiek**. Tieto sú vkladané do prúdov; každý prúd môže mať dĺžku maximálne 64kB. Jednotlivé značky majú svoj definovaný význam. Pri steganografickej technike, ktorá vkladá tajnú správu do štruktúry nosiča sa využíva značka `EOI` (*End Of Image*). Táto značka definuje koniec dátovej časti, za ktorou sa už nenachádzajú žiadne informácie potrebné pre spracovanie obrazových údajov. Keďže EOI ukončuje spracovanie toku údajov, žiadna aplikácia čítajúca `JPEG` súbor nekontroluje prítomnosť ďalších údajov v súbore. Túto vlastnosť je možné využiť na vloženie tajnej správy na koniec `JPEG` súboru za značku `EOI`. Hlavným problémom tejto techniky je jej jednoduchá detegovateľnosť a malá robustnosť. Kontrolou prítomnosti údajov za značkou `EOI` je možné okamžite zistiť prítomnosť nadbytočných (podozrivých) údajov. Malá robustnosť správy je spôsobená tým, že pri konverzii `JPEG` súboru (alebo jeho opätovnom uložení pri nejakej zmene) sa tajná informácia stratí, pretože aplikácie pracujúce s formátom `JPEG` ignorujú údaje za značkou `EOI`.

Steganografia JPEG využívajúca modifikáciu DCT koeficientov

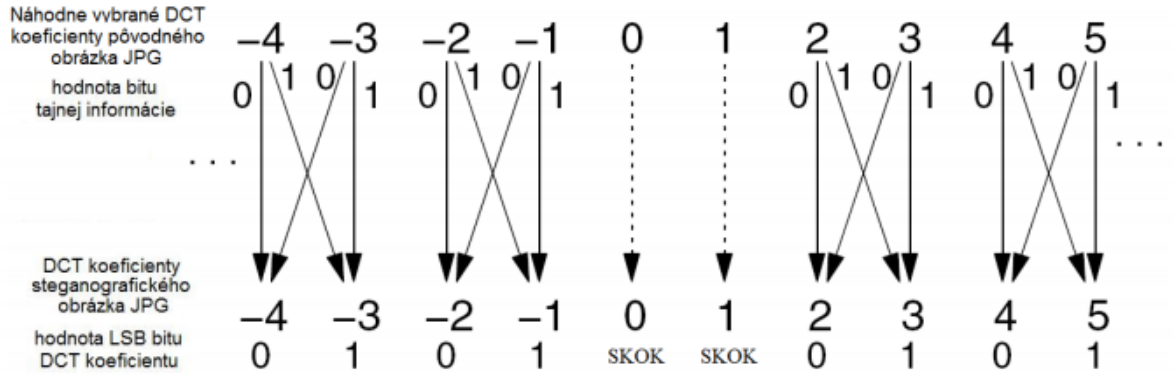
Steganografia v obrazových súborov využíva redundantné informácie, ktorých zmenu nie je možné ľudským okom spozorovať. Keďže `JPEG` kompresia je stratová (odstraňuje redundantné informácie), dlho sa myslelo, že nie je možné využiť reprezentáciu obrazu v `JPEG` kompresii na steganografické účely.

Počas procesu kvantizácie je nutné koeficienty zaokrúhľovať na celé čísla, čím sa vnáša malá (nepozorovateľná) nepresnosť do spätného zobrazenia obrázka pri jeho dekódovaní. Táto vlastnosť umožňuje využiť zmenu koeficientov o istú hodnotu, čím je možné kódovať tajnú informáciu. Najjednoduchšou metódou vloženia správy je modifikácia najmenej významných bitov koeficientov diskrétnej transformácie (DCT) po kvantizácii pred procesom Huffmanovho kódovania. Jestvuje niekoľko variácií tejto techniky:

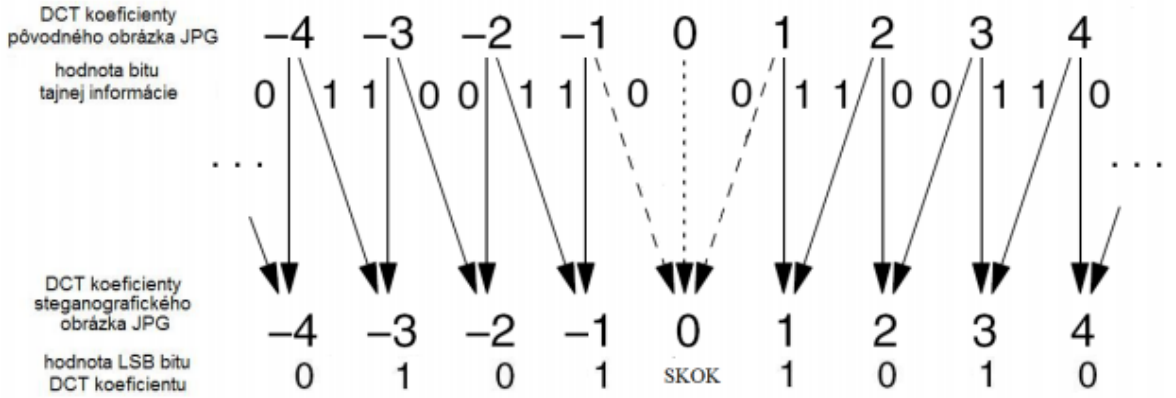
1. Sekvenčné vkladanie mení najmenej významné bity DCT koeficientov idúcich postupne za sebou. Toto vkladanie je možné odhaliť pomocou jednoduchého chí-kvadrát testu.

2. Znáhodnené vkladanie modifikuje sekvenčný algoritmus tak, že sa tajná správa nevkladá do postupnosti za sebou idúcich DCT, ale pomocou generátora náhodných čísel sa vyberajú koeficienty, ktoré budú použité pre vloženie tajnej správy.

Okrem delenia podľa výberu postupnosti DCT koeficientov použitých na tajnú správu jestvuje delenie algoritmov podľa kódovania tajnej správy:



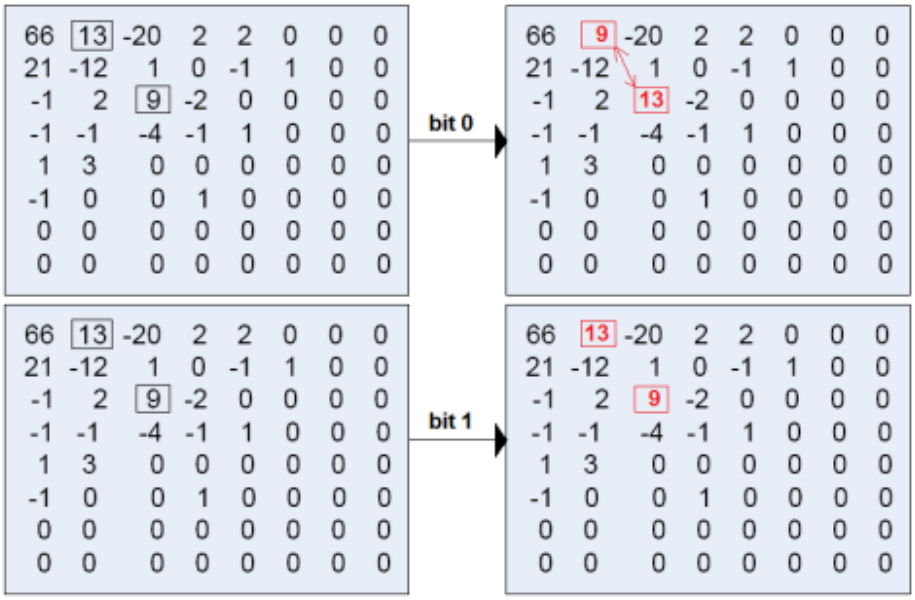
Obr. 2.1: Schéma nahrádzania LSB DCT koeficientov bitmi tajnej správy [56].



Obr. 2.2: Schéma dekrementovania LSB DCT koeficientov bitmi tajnej správy [56].

Nahrádzanie najmenej významného bitu DCT bitom tajnej správy (viď obrázok č. 2.1). Táto technika mení rozdelenie hodnôt najmenej významného bitu, preto je ľahké ju detegovať. Ak by mala tajná správa podobné rozdelenie ako hodnoty LSB, bolo by možné túto techniku použiť. Ako vidno z obrázka, dvojica DCT koeficientov (párny a nepárny) formuje tzv. navzájom zviazaný pár. Nulové DCT koeficienty nie je možné použiť z toho dôvodu, že matica DCT koeficientov má sústredené nenulové prvky v ľavom hornom rohu. Všetky ostatné prvky sú nulové. Zmenou Nulových prvkov na nenulové by sa jednak stratil význam sústredenia signálu na jednom mieste pomocou diskrétnej kosínusovej transformácie, a navyše by matica vyzerala podozrivo.

Modifikovanie absolútnej hodnoty DCT koeficientu je modifikáciou kódovania oproti predošlej techniky v tom zmysle, že v prípade nezhody najmenej významného bitu DCT koeficientu s bitom vkladanej správy sa zníži absolútna hodnota DCT koeficientu. Rozdiel je možno vidieť na obrázku č. 2.2. Pri tejto technike sa nevkladajú informácie do nulových DCT koeficientov, pretože by nebolo možné jednoznačne dekódovať správu: nebolo by možné odlíšiť, ktoré nulové DCT koeficienty sú pôvodné, a ktoré vznikli zmenou DCT koeficientu s absolútnou hodnotou 1. Keďže sa koeficienty s ab-



Obr. 2.3: Ukážka kódovania informačného bitu 0 alebo 1 pomocou techniky modulácie dvojice DCT koeficientov vyznačených štvorcami na pozíciách (1,2) a (3,3) [56].

V tomto prípade bolo dohodnuté kódovanie nulového bitu menšou hodnotou prvého koeficientu (na pozícii (1,2)). V opačnom prípade sa kóduje jednotkový bit tajnej správy.

> solútnou hodnotou 1 nemôžu použiť na kódovanie nulových bitov tajnej správy, musia sa použiť iné DCT koeficienty (v absolútnej hodnote väčšie než 1). Týmto kódovaním sa do výslednej postupnosti vnáša nerovnomernosť (oproti pôvodnému štatistickému rozdeleniu), ktorá umožňuje pozitívnu stegoanalýzu.

Steganografia JPEG využívajúca moduláciu DCT koeficientov

Oproti predošlým metódam (ktoré modifikujú najmenej významný bit DCT koeficientov) sa pri tejto metóde využíva modulácia páru DCT koeficientov. Na vloženie jedného bitu tajnej informácie sa používa jediný pár DCT koeficientov v bloku (zvyčajne veľkosti 8x8). Tým sa kapacita tejto metódy znižuje vzhľadom na predošlú. Avšak výhodou tejto metódy je jej nedetegovateľnosť, pretože pri vkladaní jedného bitu informácie vymieňa hodnoty dvoch DCT koficientov navzájom v jednom bloku. Technika modulácie spočíva na porovnaní hodnôt dvoch zvolených DCT koeficientov. Podľa ich zhody, alebo rozdielu (prvý menší než druhý alebo opačne) je možné kódovať jeden bit tajnej informácie (viď obr. 2.3).

Kódovanie oblasťami

Táto technika sa používa na ochranu pred zničením tajnej informácie manipuláciou s obrazovou informáciou. Vnáša vysokú mieru redundancie, následkom čoho má vložená informácia zväčšenú odolnosť voči narušeniu.

Z dostupných farebných bodov obrázka sa vyberú dve skupiny. Intenzita bodov prvej skupiny sa o nejakú hodnotu zvýši, kým intenzita bodov druhej skupiny sa zníži. Rozdielom

intenzít oboch skupín je možné zakódovať aspoň jeden bit informácie. Nevýhodou tohto prístupu je malé množstvo informácie, ktoré je možné zakódovať. Jestvuje mnoho variácií tejto techniky, ktoré sa snažia toto obmedzenie zmenšiť. Napríklad rozdelením obrazu na viac častí — v každej časti je možné kódovať aspoň jeden bit informácie.

V prostredí, kde sa predpokladá modifikácia nosiča tajnej správy, je vhodné použiť tento prístup, pretože na zničenie tajnej informácie by bolo potrebné zmeniť hodnotu jasu veľkého počtu obrazových bodov. Tento prístup kódovania správy je nezávislý od reprezentácie obrazu a má veľkú odolnosť voči narušeniu aj v prípade stratovej kompresie.

Kapitola 3

Steganografia vo videu

V roku 1990 bol schválený video štandard H.261, určený pre použitie v prenosovom toku ISDN. Bol jedným z prvých štandardov, ktoré využívali kompresný algoritmus JPEG. Kodér H.261 je možné používať pre služby videotelefónu a videokonferencie. Štandard MPEG-1 bol v konečnej podobe schválený v roku 1991. Bol optimalizovaný vzhľadom na prenos video signálu s rozlíšením 352x288 obrazových bodov pre televíznu normu PAL (25 obrázkov za sekundu) a 352x240 bodov pre normu NTSC (30 obrázkov za sekundu). Nasledovník štandardu MPEG-1 s označením MPEG-2 bol predstavený v roku 1994. Bol navrhnutý pre šírku pásma od 4 do 9 Mbit/s (digitálna televízia).

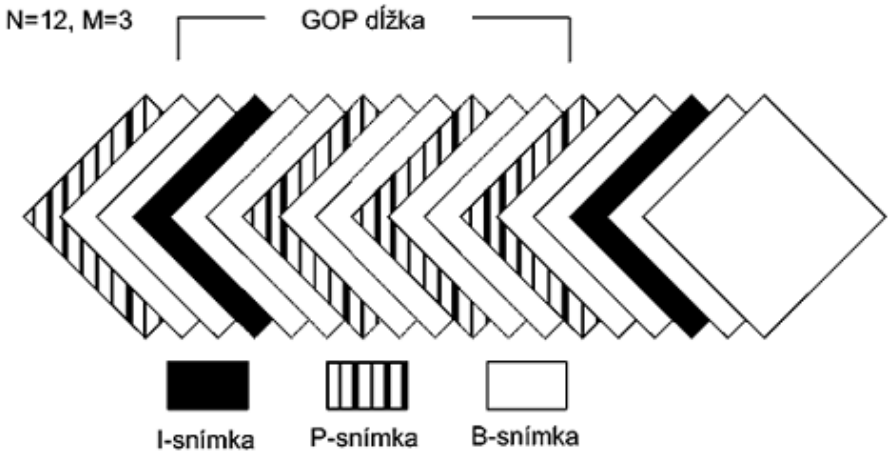
Základná schéma kodérov spomenutých štandardov je približne rovnaká. Najprv sa urobí transformácia obrazových údajov do modelu YC_bC_r. V tomto modeli je oddelená jasová (Y) zložka od obrazových (C_b pre modrú farbu a C_r pre červenú; informácie pre zelenú zložku je možné vypočítať z týchto dvoch). Dôvodom transformácie je skutočnosť, že ľudské oko je viac citlivé na jas než na farbu. Preto je potrebné jasovú zložku spracovávať s väčšou presnosťou než obrazovú. Naopak, obrazovú zložku je možné redukovať bez pozorovateľnej degradácie obrazu.

Spracovanie snímok sa môže robiť v jednom z dvoch základných módov: *inter* alebo *intra*. Pri inter móde sa ďalej spracúva rozdiel dvoch snímok (po sebe idúcich alebo rozdiel vzhľadom na referenčnú). Snímky kódované v `inter` móde sa označujú symbolom `P` alebo `B` v závislosti od toho, či sa jedná o doprednú (`P`) alebo spätnú (`B`) predikciu. V `intra` móde je každý obrázok spracovaný nezávisle od ostatných (tieto snímky sa zvyknú označovať symbolom `I`).

Pred konečným kódovaním sa snímka (alebo rozdiel snímok) rozdelí na bloky (zvyčajne veľkosti 8×8 bodov), ktoré sa spracujú diskrétnou kosínusovou transformáciou. Bloky 8×8 bodov sú vo všetkých štandardoch najmenšou údajovou jednotkou.

Skupina za sebou idúcich snímok sa označuje ako GOP (*Group of Pictures*) (toto označenie má jedine v štandarde H.261 iný význam). Skupina je tvorená začiatočnou `I` snímkou. Za ňou môžu ísť `P` alebo `B` snímky. Počet snímok medzi dvoma `I` snímkami určuje dĺžku GOP-u. V štandarde sa táto veličina označuje symbolom N. Parameter M určuje vzdialenosť snímky `P` od `I`. Ukážku skupiny možno vidieť na obrázku 3.1.

`P` **snímky:** Na základe počiatočného obrázka typu `I` v GOP-e môže kodér odhadovať nasledujúci obrázok. Obrázky odhadované dopredu sa označujú ako tzv. `P` snímky. Tento typ snímok môže byť zdrojom pre nasledujúce `P` snímky. Ak budeme uvažovať naprí-



Obr. 3.1: Ukážka GOP-u s parametrami N=12 a M=3 [63].

klad postupnosť typov snímok `I,P,P,P,P,P,I,P,P,P,P...`, každý `P` snímok je odhadnutý na základe predošlej snímky.

`B` **snímky:** V štandardoch MPEG má kodér možnosť zvoliť okrem doprednej predikcie aj snímky, ktoré sa určujú spätnou predikciou. Spätná predikcia bola zavedená z toho dôvodu, že pri doprednej predikcii nie je možné správne zobrazovať pozadie pohybujúceho sa predmetu. `B` snímky sa tvoria pomocou obojsmernej interpolačnej predikcie. Najprv sa určí napríklad dopredná predikcia, potom spätná. Kodér zvolí, v akom pomere sa spraví priemer oboch predikcií a tento výsledok sa stane základom pre kompresiu podľa pravidiel kompresie `P` snímok. Počet použitých `B` snímok v GOP-e nie je nijak štandardom obmedzený. Býva však zvykom mnohých kodérov, že používajú dve `B` snímky medzi ostatnými typmi, napríklad: `I,B,B,P,B,B,P...` Keďže interpolácia obrazu `B` snímkami vnáša najviac chýb (spomedzi ostatných typov snímok) do výsledného obrazu, použitie dvoch `B` snímok je ideálnou voľbou medzi kompresiou a výslednou kvalitou.

Najväčšou výhodou použitia `B` snímok je efektivita kódovania. `B` snímky nie je možné použiť pre predikciu nasledujúcich (alebo predošlých) snímok. Tým je zaručené, že chyby spôsobené týmto kódovaním obrazu nebudú šírené ďalej. Hlavnou nevýhodou je zdvojnásobenie pamäťových nárokov, pretože treba v pamäti uchovávať dve snímky — referenčnú spätne i dopredu.

3.1 Vkladanie údajov na základe výberu kvantizátora

Pri vkladaní dát založenom na výbere kvantizátora rozhoduje vkladaný bit o výbere jedného z dvoch kvantizátorov Q_0, Q_1. Prepokladajme, že na kvantizáciu DCT koeficientov bloku sa používa kvantizátor s konštantným krokom Δ. Nech nepárne hodnoty intenzít rekonštruovaných bodov bloku reprezentujú ukrytý jednotkový bit a naopak párne násobky Δ sú použité na vloženie nulového bitu. Potom na základe hodnoty vkladaného bitu vyberáme jeden z dvoch uniformných (mapované intervaly sú rovnakej dĺžky) kvantizátorov, ktorých krok (veľkosť mapovaného intervalu) má hodnotu 2Δ.

V práci [12] boli navrhnuté dva postupy vkladania dát postavené na výbere kvantizátora:

- schéma prahovej entropie,
- selektívne vkladanie.

Tieto metódy boli neskôr použité pri vkladaní údajov do MPEG formátu.

Schéma prahovej entropie

Pri schéme prahovej entropie (ET) sa využíva energiu bloku na určenie, či sa do daného bloku majú ukryť dáta alebo nie. Do bloku sa vkladajú dáta, keď entropia bloku presahuje vopred stanovenú hranicu. Vkladanie dát prebieha nasledovným spôsobom. Podobne ako pri JPEG-u sa aplikuje DCT na každý z neprekrývajúcich sa blokov. Po kvantizácii príslušných DCT koeficientov nasleduje výpočet entropie bloku. Nech symbol $F_{i,j} = DCT2(P)$ označuje príslušný DCT koeficient bodu $P_{i,j}$ so súradnicami $i, j \in \{0,\dots,7\}$ v rámci bloku. Teda $F_{i,j} = DCT2(P_{i,j})$, kde DCT2 symbolizuje dvojrozmernú DCT. Ďalej označme symbolom $a_{i,j}$ intezitu („farbu“) bodu $P_{i,j}$. Pre výpočet energie bloku používame nasledovný vzťah:

$$E = \sum_{i,j} |F_{i,j}|^2,$$

kde $i, j \in \{0,\dots,7\}$ okrem páru $(i, j) \neq (0,0)$. DC koeficient bloku sa nepoužíva ani na výpočet entropie, ani na vkladanie dát. Dôvodom je skutočnosť, že JPEG používa prediktívne kódovanie DC koeficientov a teda každé vloženie, ktoré zmení DC koeficienty bloku by spôsobilo zmenu DC koeficientov v ďalších blokoch.

Ako sme už na začiatku spomenuli, pre vkladanie dát sa používajú len tie bloky, ktorých energia E presiahne určitú vopred stanovenú hranicu. Vybrané bloky sú následne podelené JPEG kvantizačnou maticou pre zvolený faktor kvality QF. Ten určuje maximálny stupeň JPEG kompresie, pre ktorý zostanú zachované ukrývané dáta.

Označme $M_{i,j}^{QF}$ prvky kvantizačnej matice pre daný QF pre $i, j \in \{0,\dots,7\}$. Po delení bloku kvantizačnou maticou dostávame nové koeficienty $\tilde{F}_{i,j}$, ktoré sa použijú na vkladanie dát. Tie sa teda vypočítajú podľa vzťahu

$$\tilde{F}_{i,j} = \frac{\tilde{c}_{i,j}}{M_{i,j}^{QF}}.$$

Z týchto novozískaných koeficientov sa cik–cak prechodom (podobne ako pri JPEG) vytvorí vektor $\tilde{C} = (\tilde{c}_0,\dots,\tilde{c}_{63})$ so 64 zložkami. Dáta sa vkladajú do prvých n AC koeficientov, teda na vkladanie sa použijú len AC koeficienty nízkej frekvencie. Jednotlivé bity sa vkladajú pomocou voľby kvantizátora. Nech $B = b_0, b_1,\dots, b_n$ pre $b_k \in \{0,1\}$ predstavuje sekvenciu bitov, ktoré chceme do obrázka vložiť. Bit, ktorý chceme vložiť, určuje jeden z dvoch možných kvantizátorov. Teda bit $b_0 = 0$ (resp. $b_0 = 1$) vložíme tak, že na prvý blok s požadovanou entropiou aplikujeme Q_0 (resp. Q_1) kvantizátor. Bit b_1 vkladáme do ďalšieho prahovej entropii vyhovujúceho bloku opäť pomocou použitia Q_0 alebo Q_1 kvantizátora, atď.

Po vložení bitu b_l do AC koeficientov l-teho bloku s požadovanou entropiou dostávame vektor nových hodnôt AC koeficientov $\tilde{D} = \{\tilde{d}_0,\dots,\tilde{d}_{63}\}$, ktorého zložky možno vypočítať podľa vzťahu

$$\tilde{d}_k = \begin{cases} Q_{b_l}(\tilde{c}_k), & \text{pre} \quad 1 \leq k \leq n, \\ \tilde{c}_k & \text{pre zvyšné } k. \end{cases}$$

V ďalšom kroku je potrebné vypočítať nové hodnoty intenzít (farieb) $f'(i,j)$ bodov bloku, ktorými po JPEG komprimácii dostaneme hodnoty $\tilde{d}_k$ kvantizovaných DCT koeficientov. Tie získame tak, že z vektora $\tilde{D}$ spätne vytvoríme maticu DCT koeficientov. Tú prenásobíme JPEG-kvantizačnou maticou a aplikujeme inverznú DCT. Koeficienty s nízkou frekvenciou sa používajú na vkladanie len v blokoch, ktoré vyhovujú požadovanej entropii. Dôvodom je, že vkladanie do týchto koeficientov spôsobí minimálne skreslenie vzhľadom na JPEG kvantizáciu.

Vo všeobecnosti kompresia (kvantizácia DCT koeficientov) znižuje celkovú entropiu bloku. Preto v „nekódovej“ verzii schémy je nevyhnutné kontrolovať, či výsledná entropia bloku (po vložení) je stále vyššia ako požadovaná prahová hodnota. Pokiaľ entropia pôvodného bloku presahuje prahovú hodnotu, avšak po vložení dát sa táto hodnota zníži pod prah, ponecháva sa tento blok nezmenený a informačný bit sa vkladá do nasledujúceho vyhovujúceho bloku.

Poznámka 3. *Tento prístup vedie ku problémom pri dekódovaní. Ako je uvedené ďalej, dekodér podľa entropie bloku zisťuje, či sa v ňom nachádzajú údaje. Ak je v procese vkladania blok, ktorý spĺňa kritérium entropie (pred vložením údajov, avšak po vložení už nie) preskočený, dekodér nemá dostatočnú informáciu o tom, aby rozhodol, ktorý bol a ktorý nebol preskočený. Preto navrhujeme takú úpravu algoritmu, pri ktorej sa informačný bit kóduje pomocou bloku, ktorý vložením bitu zníži svoju entropiu pod prahovú hodnotu, a zároveň sa tento istý informačný bit bude vkladať do nasledujúceho vhodného bloku.*

Pri dekódovaní počíta dekodér entropiu každého bloku, aby zistil, v ktorých blokoch sú ukryté dáta. V uvedenej schéme sú dekodéru aj kodéru známe dva parametre a to: prahová entropia bloku a množina koeficientov bloku používaných na vkladanie dát. Uvedené parametre sú nezávislé na obrázku. Ich hodnota je priamo určená len faktorom kvality.

Selektívne vkladanie

Pri selektívnom vkladaní (SEC) vyhodnocujeme nezávisle každý DCT koeficient bloku, či do neho vložíme informáciu alebo nie. Cieľom je určiť tie koeficienty, ktorých zmena spôsobí z hľadiska ľudského vnímania minimálne skreslenie obrázka. Postup pri selektívnom vkladaní je pre prvú časť vkladania rovnaký ako pri ET schéme. Aplikáciou DCT2 sa získajú hodnoty DCT koeficientov, vydelia sa JPEG kvantizačou maticou, ktorej výsledkom je rovnako ako pri ET vektor hodnôt $\tilde{C} = (\tilde{c}_0, \ldots, \tilde{c}_{63})$. Rovnako ako pri ET schéme, aj tu sa vopred stanoví hodnota parametra n. Ten určuje, ktoré z $\tilde{c}_k$ prichádzajú do úvahy pre vloženie informácií. Konkrétne sa jedná o prvých n AC koeficientov, teda vloženie môže byť uskutočnené do koeficientov $\tilde{c}_k$, pre $k \in \{1, \ldots, n\}$. V ďalšom kroku sa tieto hodnoty kvantizujú JPEG kvantizátorom Q_I, ktorý ich namapuje na najbližšie celé čísla. Z nich sa následne vypočíta absolútna hodnota r_k, ktorej veľkosť určí, či sa do $\tilde{c}_k$ bude vkladať informácia alebo nie. Teda pre

$$r_k = |Q_I(\tilde{c}_k)|, \quad 1 \le k \le n$$

sa do $\tilde{c}_k$ vkladá informácia, ak hodnota r_k je väčšia ako vopred stanovený prah t. Informácia sa rovnako ako pri ET schéme vkladá výberom jedného z dvoch kvantizátorov $Q_0(.), Q_1(.)$. Takto sa získa vektor $\tilde{D} = \{\tilde{d}_0, \ldots, \tilde{d}_{63}\}$, kde hodnoty $\tilde{d}_k$ sa získajú podľa vzorca:

$$\tilde{d}_k = \begin{cases} Q_{b_l}(\tilde{c}_k), & ak \quad 1 \le k \le n, \quad \text{a} \quad r_k > t \\ r_k, & r_k = t \\ \tilde{c}_k & \text{v ostatných prípadoch} \end{cases}$$

Po získaní vektora $\tilde{D}$ sa opäť ako pri ET schéme vypočítajú také nové hodnoty $f'(i,j)$ intenzít bodov bloku, aby sa po JPEG kompresii a kvantizácii získal vektor $\tilde{D}$.

Pri selektívnom vkladaní je nutné testovať, či absolútna hodnota koeficientov leží medzi t a $t+1$. Ak by totiž kvantizovaná hodnota $Q_{b_l}(\tilde{c}_k)$ bola rovná prahu t, dekodér by nevedel rozhodnúť, či nebol tento koeficient vybraný pre nízku absolútnu hodnotu alebo bol l-tý bit b_l použitý na vloženie.

Najjednoduchšou verziou selektívneho vkladania je verzia s nulovým prahom $t = 0$, kde sa použijú na vloženie tie koeficienty, ktoré sa nekvantizovali na nulu. To umožňuje vložiť vysoké množstvo bitov s relatívne malým skreslením obrazu, ktoré sa podobná JPEG komprimovaným obrázkom.

Intuitívne sa to dá vysvetliť nasledovne: po JPEG-kvantizovaní sa množstvo DCT koeficientov namapuje na nulu. Vloženie bitu 1 do takýchto koeficientov spôsobí veľké skreslenie s ohľadom na pôvodnú hodnotu DCT koeficientov. Preto je nutné vylúčiť vkladanie do nulových DCT koeficientov. Ako rastie hodnota prahu, stále menej a menej koeficientov vyhovuje kritériu na vloženie. To indukuje stále kvalitnejší obraz. Pre prah $t \geq 2$ už prakticky človek nedokáže vizuálne odlíšiť pôvodný obraz a obraz po vložení pomocou SEC. Na predstavu pre obraz s rozmermi 512×512 bodov a prah $t = 0$ možno vložiť okolo 2800 bitov, ktoré sa zachovajú JPEG kompresiou s $QF = 25$ a predsa obraz je takmer neodlíšiteľný od pôvodného.

Pri selektívnom vkladaní je v porovnaní s ET schémou väčšia kontrola nad tým, kam sa informačné bity vkladajú. Preto metóda SEC umožňuje ukryť rovnaké množstvo dát s menším celkovým skreslením obrazu ako ET schéma. Ďalšou veľkou výhodou SEC je, že sa dá priamo určiť množstvo dát, ktoré sa dajú do média vložiť. Z tohoto dôvodu sa tejto metóde dala prednosť pri jej zovšeobecnení na formát MPEG-2.

3.2 Adaptívna metóda ukrývania údajov do videoformátu MPEG-2

V [70] autori navrhli spôsob, ako použiť metódu selektívneho vkladania dát do DCT koeficientov nepohyblivých obrázkov popísanú v [75] pri vkladaní do videoformátu MPEG-2. V tejto časti ju popisujeme.

Na začiatku sa vstupné video pomocou dekodéra dekomprimuje a rozloží na postupnosť snímok (obrázkov). Dáta sa vkladajú len do jasovej zložky (Y). Koeficienty sa podelia JPEG-kvantizačnou maticou zvoleného faktora kvality. Pri kvantizácii sa používa uniformný kvantizátor s krokom veľkosti Δ. Pri vkladaní sa používa schéma selektívneho vkladania do koeficientov popísaná vyššie. Správa sa vkladá len do DCT koeficientov nízkych frekvencií, ktorých hodnota je väčšia ako vopred zvolená prahová hodnota. Bity správy sú vkladané výberom jedného z dvoch kvantizátorov, oba s krokom veľkosti 2Δ. Nepárne násobky kroku Δ sa používajú na vloženie jednotkového bitu a párne na vloženie nulového bitu.

Pri dekódovaní používa dekodér pri určovaní, či bola do príslušných DCT koeficientov ukrytá správa, rovnaké kritériá ako kodér. Problémom pri dekódovaní môže byť šum vnesený prenosovým kanálom. Ten môže spôsobiť dva druhy chýb: „vloženie“ (dekodér nesprávne určí, že je v DCT koeficiente ukrytá informácia) a „vymazanie“ (dekodér nesprávne určí, že v danom DCT koeficiente nie je ukrytá informácia). Aby sa predišlo uvedeným dvom typom chýb, autori navrhli použitie samoopravných kódov. Z dôvodu jednoduchej implementácie autori používajú RA-kód, kde sa každý bit q-krát opakuje.

Schéma adaptívneho ukrývania správ do videoformátu MPEG-2

Pri vkladaní dát do obrázkov môžeme meniť nasledovné štyri parametre:

- kapacitu n bloku (počet AC koeficientov n bodov bloku, do ktorých sa vkladá správa),
- faktor q (určuje odolnosť schémy na možné útoky) RA kódu, kde sa každý bit kóduje postupnosťou q-bitov s rovnakou hodnotou,
- veľkosť Δ kvantizovaných intervalov,
- faktor kvality QF použitý pri kvantizácii DCT koeficientov.

Počet bitov, ktoré možno vložiť do bloku, závisí na kapacite n a počte opakovaní q každého bitu. Teda do bloku možno vložiť n/q bitov správy.

Popíšme si teraz bližšie charakteristiku schémy:

1. Formát MPEG-2 využíva tri typy snímok: `I`, `P` a `B`. Vďaka obojsmernej predikcii sú B-snímky reprezentované minimálnym množstvom bitov[1]. Naopak, `I` snímky sú reprezentované najväčším počtom bitov. Prirodzene je teda skreslenie po vložení do `B` snímok maximálne a minimálne pri `I` snímkach. Z toho dôvodu je možné vložiť viac dát do `I` snímok v porovnaní s `B` snímkami.

2. Pri videoformáte MPEG-2 je nutné riadiť množstvo spracovávaných dát. Za týmto účelom autori používajú schému priamo z MPEG-2, ktorá úzko súvisí s kvantizačným parametrom. Z práce [74] vieme, že kvantizačný parameter pre makroblok závisí od prenosovej rýchlosti, frekvencie, s akou sa menia snímky a priestorovej aktivity makrobloku. Nízke hodnoty parametra indikujú väčšiu aktivitu bloku a kvalitnejšiu kvantizáciu. Výpočet priemerného kvantizačného parametra (cez všetky makrobloky snímky) nám poskytuje odhad počtu bitov potrebných na reprezentáciu snímky kodérom. V ďalšom budeme tento priemerný kvantizačný parameter označovať v zhode s [74] ako $mquant$.

3. Aby vkladanie pomocou SEC metódy bolo odolné voči šumu, vkladáme len do tých kvantizovaných DCT koeficientov, ktoré presiahnu určitú hranicu (prah). Pre hladkú oblasť (oblasť bodov veľmi blízkej jasnosti) obrázka má väčšina AC koeficientov hodnotu blízku nule, a teda nie sú vhodné na vkladanie. Pre bloky s väčším rozptylom jasnosti je veľkosť DCT koeficientov vo všeobecnosti väčšia. Preto majú tieto väčšiu kapacitu pre vkladanie. Z tohoto dôvodu sa využívajú na vkladanie správy práve tieto bloky. Snímky s takýmito blokmi majú nižšiu hodnotu $mquant$.

Kvantizačný parameter formátu MPEG-2

Pri formáte MPEG-2 je potrebné vysvetliť, ako sa vypočíta kvantizačný parameter pre daný makroblok. MPEG-2 používa adaptívnu schému kontroly rýchlosti: na základe typu snímky a bitov vyhradených pre snímku sa mení referenčný kvantizačný parameter Q_j j-teho makrobloku. Zo štandardu MPEG-2 vieme, že hodnota parametra Q_j sa vypočíta pomocou vzorca

$$Q_j = \left(\frac{31 * d_j}{r}\right),$$

[1] minimalizácia bitov je zabezpečená medzisnímkovým kódovaním a absenciou pohybového vektora

kde r predstavuje „reakčný parameter“, ktorého hodnota je daná vzorcom:

$$r = \frac{2 * bit\_rate}{picture\_rate}.$$

Premenná bit_rate predstavuje prenosovú rýchlosť video signálu (napr. 1.5 MBps, 4 MBps, atď.), $picture\_rate$ snímkovú frekvenciu (napr. 25 snímok/sekundu) a d_j je miera naplnenia virtuálneho zásobníka makrobloku.

Vo formáte MPEG-2 sa menia hodnoty d_j v závislosti od typu snímky `I`, `B`, `P` a v závislosti od priestorovej aktivity makrobloku. V každom makrobloku sa pomocou hodnoty Q_j vypočíta $mquant_j$, ktorý sa následne použije pri kvantizácii celého makrobloku. Miera priestorovej aktivity act_j pre j-ty makroblok sa vypočíta podľa MPEG-2 z ôsmich jasových podblokov makrobloku a pôvodných hodnôt jasu pomocou vzťahov:

$$act_j = 1 + \min(vblk_1, \ldots, vblk_8)$$

$$vblk_n = \frac{1}{64} \sum_{k=1}^{64} (P_k^n - P_{mean_n})^2$$

$$P_{mean_n} = \frac{1}{64} \sum_{k=1}^{64} P_k^n.$$

V uvedených vzťahoch predstavuje P_k hodnotu intenzity jasu v n-tom bloku, act_j hodnotu aktivity j-teho makrobloku.

Nech avg_act predstavuje priemernú aktivitu. Potom normalizovaná hodnota pre aktivity jednotlivých makroblokov je daná vzťahom

$$N_{act} = \frac{2 * act + avg\_act}{act + 2 * avg\_act}.$$

Pomocou normalizovaných aktivít možno vypočítať kvantizačný parameter $mquant_j$ ako

$$mquant_j = Q_j * N_{act_j}.$$

Výpočet kapacity média

Uvažujme nasledovnú situáciu: Nech X predstavuje postupnosť bitov správy, ktorú vkladáme do snímky. Táto prejde cez kanálový kodér, ktorý q-krát „naklonuje“ každý bit správy. Do snímky sa teda vkladá postupnosť Y, kde je každý bit z X q-krát zopakovaný. Postupnosť Y sa následne vkladá do DCT koeficientov. Ak spadá výsledná hodnota DCT koeficientov do intervalu $< -\Delta/2, \Delta/2 >$, hovoríme o „vymazaní“ (označujeme ho ako e). Teda v závislosti na hodnote DCT koeficientu hovoríme, že sa vložil bit $1,0$ alebo e. Označme symbolom Z túto postupnosť vkladaných bitov. Po vložení zakódujeme snímky do MPEG-2 formátu a prenesieme ich.

Po dekódovaní MPEG-2 videa získame bity správy z kvantizovaných DCT koeficientov pre každú snímku. Používame nasledovné pravidlo: z koeficientu extrahujme bit 0 alebo 1 podľa toho, či sa jedná o párny alebo nepárny násobok kroku Δ kvantizátora. Označme výstupnú postupnosť bitov $0,1$ a e dekodéra ako Z'. V závislosti na postupnosti Z' aplikujeme RA-q dekódovanie a obdržíme rekonštruovanú postupnosť X', ktorá sa môže od pôvodnej postupnosti X líšiť. Nás zaujíma, ako nastaviť parametre schémy tak, aby sa X a X' líšilo čo

najmenej. Z experimentov získame matice pravdepodobností prechodu postupnosti Y do Z a Z do Z'. Matice závisia na zvolených parametroch (n,Δ, a QF). Pre výpočet kapacity cap jednej snímky je potrebné maximalizovať veľkosť vzájomnej informácie medzi vstupnou postupnosťou Y a výstupnou postupnosťou Z':

$$cap = \max_{p(y)} I(Y,Z),$$

$$I(Y,Z) = \sum_{y\in 0,1, z'\in 0,1,e} p(y,z') \log\left(\frac{p(y|z')}{p(y)}\right).$$

Pri zafixovaní hodnôt n,Δ a QF možno vypočítať kapacitu cap snímky využitím predchádzajúceho vzťahu. Pomocou nej možno ďalej stanoviť ideálnu hodnotu q_{eff} parametra RA kódu. Tá má hodnotu $\lceil \frac{1}{cap} \rceil$.

3.3 Technika ukrývania pomocou VLC stromov

V prípade použitia steganografických techník pre komprimované video musí byť správa vkladaná do entropicky kódovanej časti videa. Táto časť je zložená z kódov s premenlivou dĺžkou (VLC, angl. *variable length code*), ktoré reprezentujú rozličné segmenty videa, vrátane intrakódovaných makroblokov, pohybových vektorov, atď.

Prvé použitie steganografie vo VLC bolo odvodené od známej LSB metódy [26]. Najprv sa identifikuje podmnožina VLC, ktorá reprezentuje rovnaký beh (angl. *run*), avšak líši sa v úrovni o 1. Tieto VLC sa nazývajú tzv. úroveň nesúce VLC. Následne sa bit správy porovná s LSB bitom úrovne nesúceho VLC. Ak sa tieto bity zhodujú, VLC úroveň zostane nezmenená; ak sú tieto bity rozdielne, tak bit správy vložíme do LSB úrovne nesúceho VLC. Dekodér potom len jednoducho extrahuje LSB bity úrovne nesúcich VLC. Tento algoritmus je veľmi jednoduchý a rýchly a bol úspešne implementovaný [13].

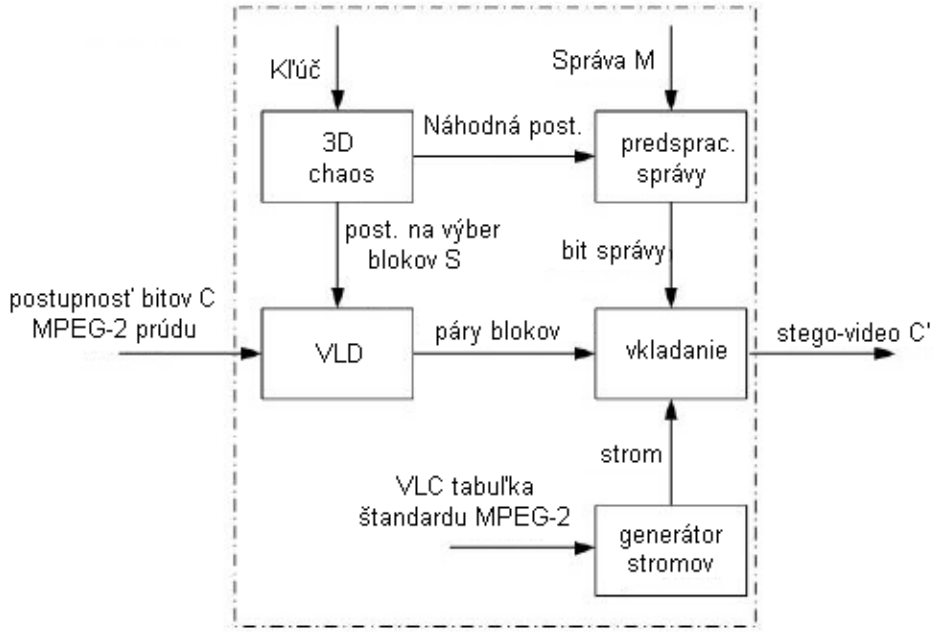
V práci [7] je použitý prístup založený na tzv. VLC mapovaní. Tento prístup vychádza z návrhu kódového priestoru.

Metóda VLC párov pre MPEG-2 steganografiu je diskutovaná v [55]. Táto metóda vyriešila problém malej veľkosti kódového priestoru, ale konštrukcia párovacieho stromu je značne náročná.

V tejto časti popisujeme techniku vkladania informácií do videa v komprimovanej oblasti, založenú na VLC párach [53]. Proces vkladania je adaptívne riadený viacerými VLC stromami, ktoré sú generované z hlavnej VLC tabuľky, špecifikovanej štandardom `ISO/IEC13818-2:1995`. Táto technika nevyžaduje žiadnu (ani len čiastocnú) dekompresiu videa.

Na obrázku 3.2 je znázornená architektúra diskutovaného steganografického systému. Ako vidno, je tvorený piatimi modulmi.

Trojrozmerné chaotické mapovanie je použité na generovanie troch pseudonáhodných číselných postupností. Dve z nich sú zaslané do VLD dekodéra a tretia vstupuje do preprocesora vkladania. VLD dekodér spracúva bitový prúd MPEG–2 a vyberá dva dátové bloky, na ktoré ukazujú náhodné postupnosti. Vybraný pár blokov (b_{1i} a b_{2i}) je zaslaný bo bloku vkladania. Spomínaná tretia náhodná postupnosť môže byť použitá na moduláciu ukrývanej správy. Po predspracovaní vstupuje utajovaná správa do bloku vkladania. Algoritmus vkladá bity utajovanej správy (m_i) do blokov b_{1i} a b_{2i} použitím VLC stromov. Najdôležitejšie časti tohto systému — generátor VLC stromov a vkladací mechanizmus — sú popísané nižšie.



Obr. 3.2: Architektúra steganografického systému založeného na VLC stromoch [53].

Generovanie VLC stromov pre MPEG–2 prúd

V MPEG štandarde je VLC kódovanie založené na kódových tabuľkách. Nech je VLC kódovacia tabuľka tvorená N kódmi označenými $V = \{v_1, v_2, \ldots, v_N\}$, kde v_i je i-ty VLC dĺžky l_i, $l_i < l_j$ a $i < j$. Steganografia pre v_i je definovaná ako preklopenie jedného alebo viacerých bitov v_i. Napr. ak je k-ty bit použitý pre steganografiu, tak potom $v_j^k = \{v_{i1}, v_{i2}, \ldots, \bar{v}_{ik}, \ldots, v_{il_i}\}$. Ak sa v_i^k dostane mimo očakávaných hodnôt, resp. mimo priestoru kódu V, tak dekodér môže „umlčať“ takýto VLC. Avšak, väčšina komprimovaných bitových tokov, napr. JPEG, používajú Huffmanovo kódovanie, takže takéto mapovania v_i buď predstavujú ďalší validný VLC alebo dochádza pri nich k porušeniu podmienky pre prefix, čo býva označované aj pojmom kolízia. Aby sa predišlo kolíziám, VLC sú mapované do kódového stromu. Kódový strom pre VLC je binárny strom, kde VLC zaberajú listové uzly. Strom je tvorený l_N úrovňami. Koreň leží v úrovni 0. Každá úroveň môže byť tvorená až 2^l uzlami. Aby bola splnená prefixová podmienka, žiaden z kódov vo V nesmie ležať na podstrome definovanom pomocou v_i, pozostávajúcom zo všetkých možných listových a vnútorných uzlov od l_i až po l_N. Takýto kódový strom môže byť použitý na identifikáciu tých VLC, ktoré môže byť jednoznačne spracované.

VLC tabuľky v štandarde MPEG–2 obsahujú 113 VLC, nerátajúc znamienkový bit. Dva VLC, ktoré majú rovnakú hodnotu behu a líšia sa iba v hodnotách pre úroveň, majú rozličnú kódovú dĺžku. Tieto VLC potom možno rozdeliť do niekoľkých skupín s rovnakými hodnotami zmeny kódovej dĺžky, ktorá sa vyskytuje pri hodnotách úrovne (plus alebo mínus 1). Definujeme preto rozšírený kódový priestor párovaním pôvodných VLC nasledovne:

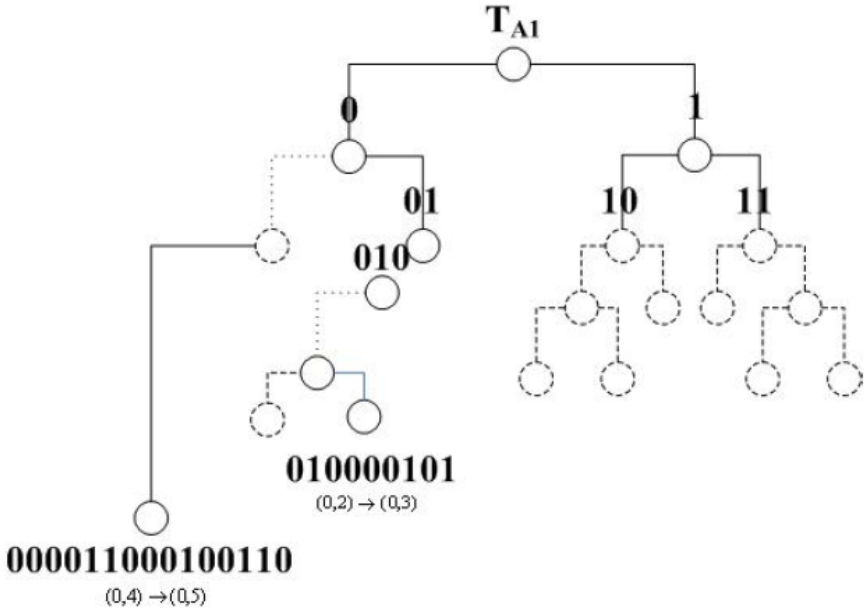
$$U = \{u_{ij}\}, \quad i, j \in \{1, 2, \ldots, N\},$$

kde $u_{ij} = \{v_i, v_j\}$, $i \neq j$ a $|v_i - v_j|$ označuje zmenu kódovej dĺžky.

Listový uzol vo VLC stromoch je kombinovaný VLC kód páru VLC. Ak dĺžka predchádzajúceho VLC bola menšia než dĺžka nového, potom takejto skupine hovoríme *pričítacia skupina* a strom generovaný touto skupinou nazývame *pričítací strom*. Ak zmena dĺžky $|v_i - v_j| = n$, takýto pričítací strom označujeme T_{An}. Odčítacie stromy sú definované obdobne. V tabuľke 3.1 je uvedených niekoľko príkladov listových uzlov v T_{A1}. V každom riadku sú uvedené dva VLC, označené pármi hodnôt (beh, úroveň), ktoré sa líšia o 1 v hodnotách úrovní a zmena dĺžky týchto VLC je 1. Použitím všetkých listových uzlov možno zostaviť VLC strom. Na obrázku 3.3 sú znázornené dva listové uzly v T_{A1}.

Tabuľka 3.1: Niekoľko príkladov listových uzlov v T_{A1}

(beh, úroveň)	(beh, úroveň)	zmena
(0,2)	(0,3)	1
(0,4)	(0,5)	1
(0,11)	(0,12)	1
(0,15)	(0,16)	1
(0,31)	(0,32)	1
(1,14)	(1,15)	1
(2,4)	(2,5)	1
(3,3)	(3,4)	1



Obr. 3.3: Dva príklady listových uzlov v T_{A1} [53].

Vkladanie a získavanie ukrývanej správy

Na zachovanie bitovej rýchlosti video sekvencie sa nastaví hodnota Δ. Δ označuje celkovú predošlú dĺžkovú zmenu. V procese vkladania ukrývanej správy môžu byť VLC stromy vyberané automaticky použitím Δ. Keď sa vkladá (ukrýva) bit správy m_i, tak sa m_i porovná so súčtom hodnôt úrovní označenými pomocou VLC v blokoch b_{1i} a b_{2i}. Ak sa zhodujú, žiadna VLC úroveň sa nemení. V opačnom prípade sa zoberú posledné dva VLC v oboch blokoch a testuje sa Δ:

- Ak $\Delta > 0$, tak budú prehľadávané T_0 a odčítacie stromy.
- Ak $\Delta < 0$, tak budú prehľadávané T_0 a pričítacie stromy.

Po nájdení vhodných dvoch VLC v stromoch sa porovnajú dve dĺžkové zmeny, pričom len menšia zmena bude ďalej spracovaná. Týmto proces vkladania končí a zmena v bitovej rýchlosti video toku je limitovaná len na veľmi nízkej úrovni.

Extrakcia ukrytej správy je veľmi jednoduchá. Pri znalosti správneho kľúča a daného trojrozmerného chaotického mapovania vie dekodér generovať rovnaké pseudonáhodné postupnosti. Vypočíta LSB zo súčtu hodnôt úrovní v dvoch vybraných blokoch (podľa pseudonáhodných postupností) a demoduláciou pomocou tretej náhodnej postupnosti sa získa bit ukrytej správy.

3.4 Bezpečná steganografia v komprimovanom videu

Digitálne video môže byť vo všeobecnosti uložené použitím dvoch formátov kódovania: nekomprimované a komprimované. Najpopulárnejším komprimovaným formátom v súčasnosti je komprimované video kompenzujúce pohyb (angl. *motion compensated compressed video*), konkrétne široko akceptovaný štandard `MPEG-x`. `MPEG-x` dosahuje kompresiu pomocou eliminácie dočasných, priestorových a štatistických redundancií použitím stratovej kompresie. Bitový tok videa pozostáva z viacerých kódov s premenlivou dĺžkou (tzv. VLC, angl. *variable length code*), ktoré reprezentujú rôzne video segmenty.

Steganografické algoritmy, ktoré nie sú aplikovateľné pre komprimovaný bitový prúd, budú vyžadovať úplnú alebo aspoň čiastočnú dekompresiu [37], [50], [12], [18]. To však predstavuje množstvo operácií navyše a časové oneskorenie, ktoré treba eliminovať. Aplikácia si však môže vyžadovať použitie steganografických techník priamo v komprimovanej oblasti. Dnes je už známych niekoľko algoritmov pre vodoznaky, aplikovaných pre komprimované video [54], [26], [55], [52], [84].

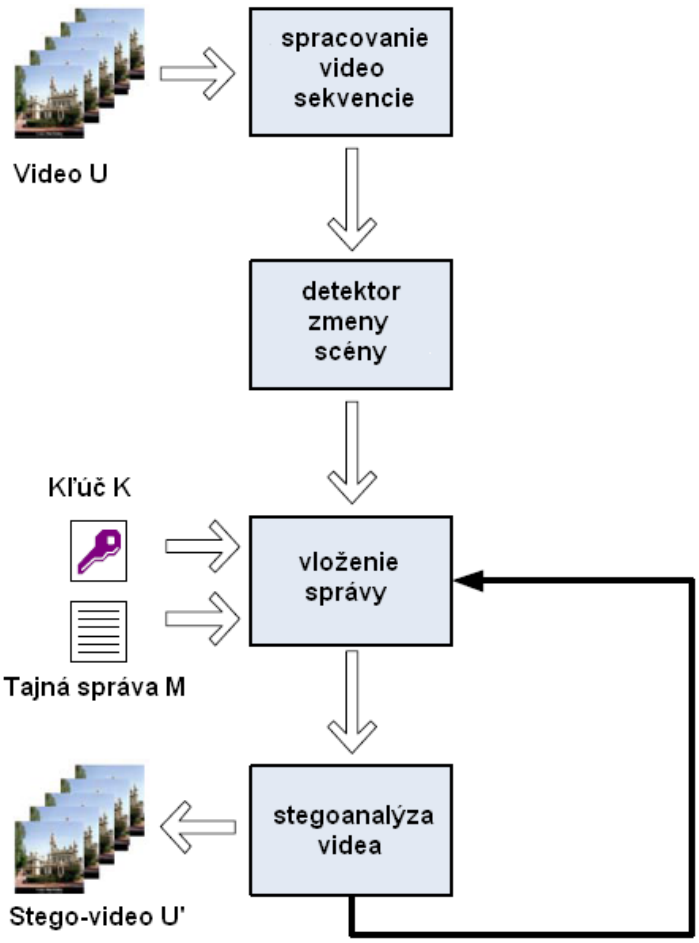
Steganografická technika, resp. utajovaná správa, ukrytá v nosiči, majú byť ťažko odhaliteľné. Ak možno detegovať existenciu ukrytej správy s pravdepodobnosťou väčšou než náhodné hádanie, považujeme takúto steganografickú techniku za chybnú, nevhodnú, resp. zlomenú. Výskum sa zatiaľ zameriaval prevažne na stegoanalýzu pre súbory s obrázkami. Jeden prístup je založený výlučne na štatistike prvého rádu a je aplikovateľný iba na idempotentné vkladanie [82], [29], [30] a [22]. Další významný smer útokov, založený na koncepte slepej stegoanalýzy, je tvorený slepými klasifikátormi [27], [35] a [3]. Klasifikátor je trénovaný, aby sa „naučil“ rozdiely medzi črtami nosičov a stego–obrázkov. Dalšie dve stegoanalytické metódy sú založené na tzv. princípe kolúzie [9] a [41].

V tejto časti diskutujeme steganografický systém z práce [51], inšpirovaný niektorými video vlastnosťami štatistickej neviditeľnosti z práce [77]. Táto metóda využíva aj proces stegoanalýzy použitý v spätnej väzbe pre vkladanie správy, ktorá vylepšuje anti–stegoanalytické vlastnosti tejto techniky.

Základný koncept techniky

Architektúra steganografického systému (obr. č. 3.4) je zložená zo štyroch základných modulov:

- parser video sekvencie,
- detektor zmien scény,
- vkladač tajnej správy a
- blok video stegoanalýzy.



Obr. 3.4: Architektúra steganografického systému v komprimovanom videu [51].

Kompresia pre video sekvencie je dosahovaná elimináciou dočasných, priestorových a štatistických redundancií použitím kompenzácie pohybu, blokovej kvantizácie pri diskrétnej kosínusovej transformácii (DCT) a Huffmanovho (angl. *run-level*) kódovania. Bitový prúd videa pozostáva z kódov s variabilnou dĺžkou (VLC). Nosič (video sekvencia) U vstupuje najskôr do bloku parsera video sekvencie, ktorý obsahuje VLD (angl. *variable length decoder*), ktorý dekóduje VLC kód označujúci rôzne segmenty videa, vrátane intrakódovaných makroblokov, DCT koeficientov, pohybových vektorov, atď.

Druhý modul — detektor zmien scény — rozdelí sekvenciu do niekoľkých pomalých, jedno-scénových video subsekvencií. Záber (snímka) vo video sekvencii je definovaný ako jeden objekt snímaný jednou kamerou na jednom mieste. Scéna je definovaná ako niekoľko následných záberov. Obrázky v jednej scéne majú podobné črty. Detektor zmeny scény má za úlohu nájsť bod zmeny scény v skupine obrázkov. V navrhnutom systéme sú v detektore zmeny scény použité DC koeficienty (kvôli požiadavke spracovania v reálnom čase). Operácie vkladania sa vykonávajú v I rámcoch, preto hľadá detektor bod zmeny scény medzi I rámcami v komprimovanom formáte.

Ako už bolo spomínané, I rámce sú v MPEG štandarde kódované vnútrosnímkovo. Teda DC obraz môžeme získať extrahovaním DC koeficientov z kódov DCT koeficientov. Nasledovný vzťah vyjadruje metódu pre porovnanie dvoch po sebe idúcich I rámcov:

$$HD(I_i, I_{i+1}) = \sum_{k=1}^{N} \frac{(H_i(k) - H_{i+1}(k))^2}{(H_i(k) + H_{i+1}(k))^2},$$

kde I_i a I_{i+1} označujú i-ty a $(i+1)$-vý I rámec, H_i a H_{i+1} sú histogramy z DC obrázkov z i-teho a $(i+1)$-vého I rámca. Ak je $HD(I_i, I_{i+1})$ extrém (angl. *peak*), tak potom dané dva I rámce sú z rozličných scén, a teda bol nájdený bod zmeny scény. Taktiež sa vypočítajú variancie $\mathrm{var}(U_i)$ jednotlivých DC obrázkov z I rámca a pošlú sa do modulu vkladania tajnej správy.

V treťom module, ktorý slúži na vkladanie tajnej správy, sa tajná správa kóduje do komprimovaného videa bez viditeľného (perceptívneho) narušenia obrazu. Tento modul je kľúčovou časťou steganografického systému a bude bližšie popísaný neskôr.

Posledný modul, stegoanalyzátor videa, je použitý na overenie bezpečnosti vloženej správy. Podľa princípu kolúzie [9], odchýlka DC a AC koeficientov v následných rámcoch je štatisticky viditeľná. Taktiež korelácia medzi rámcami v jednej scéne je použitá ako ďalšia miera dátovej zmeny. Použitím štatistík prvého a vyšších rádov ako vstupov pre umelo-inteligenčný klasifikátor, môže byť vloženie správy v stego-video sekvencii detegované. Ak stego-video neprejde testom v tomto module, musí byť upravený faktor prispôsobenia.

Výber pozície pre vkladanie

Vo VLC tabuľke v MPEG štandarde každé kódové slovo korešponduje nejakému beh-úroveň páru, označenému angl. (*run*, *level*). Počas kódovania videa sú hodnoty jednotlivých pixelov (rozmeru 8×8) v priestorovej oblasti transformované pomocou DCT. Po kvantizácii je obsah každého 8×8 bloku skenovaný tzv. cik–cak metódou. Každý skenovaný nenulový koecifient musí byť konvertovaný na (run, level) pár a následne, podľa VLC tabuľky, konvertovaný na príslušný bitový tok.

V prípade, že zvolíme (run, level) páry ako potenciálne miesta pre ukrývanie bitov správy, možno modifikovať len hodnoty úrovne (level). Podľa MPEG kódovacieho pravidla, ak sa moduluje hodnota behu, znamená to zmenu počtu predchádzajúcich núl. Takáto zmena by

viedla k vážnym dôsledkom — pozície všetkých následných nenulových koeficientov by boli posunuté. Keď potom takto posunuté koeficienty dekódujeme naspäť do priestorovej oblasti, získame rámec výrazne odlišný od pôvodného. Avšak, ak zmeníme len hodnotu úrovne, potom sa zmení iba jeho významnosť.

Ďalšou otázkou v oblasti video steganografie je výber vhodného farebného kanála pre vkladanie. Zvyčajne sa rámce video sekvencie rozdeľujú na `Y`, `Cb` a `Cr` kanály v procese MPEG kódovania. Podľa odlišných pravidiel pre farebné vzorkovanie, pomer `Y:Cb:Cr` môže byť nastavený na 4:2:2 alebo 4:2:0. Preto pre vkladanie správy je preferované použitie Y kanála.

Nakoniec je potrebné vybrať vhodný typ rámca pre vkladanie. Zvyčajne je bežné video tvorené viacerými GOP (angl. *group of pictures*). Každá GOP je tvorená jedným `I`-rámcom a niekoľkými `B`-rámcami a `P`-rámcami. Typický `I`-rámec je intrakódovaný, t.j. nereferuje na žiadne iné rámce. `P`-rámce odkazujú len na najbližší predchádzajúci `I` alebo `P`-rámec. `B`-rámce odkazujú na najbližší predchádzajúci a/alebo nasledujúci `I` alebo `P`-rámec. V navrhnutom algoritme sa správa vkladá do `I` rámcov MPEG komprimovanej video sekvencie.

Analýza vkladania v komprimovanej oblasti

V nasledujúcom texte priblížime efekt vkladania v komprimovanej oblasti na varianciu hodnôt jednotlivých obrazových bodov v priestorovej oblasti.

Nech $f(x,y)$ je hodnota bodu v priestorovej oblasti na pozícii (x,y) v `I` rámci. Nech $\Delta f(x,y)$ značí inkrement hodnoty bodu. $D(f(x,y))$ značí varianciu všetkých bodov v `I` rámci a $\Delta D(f(x,y))$ potom značí inkrement $D(f(x,y))$. Nech $F(u,v)$ je hodnota DCT koeficientu vo frekvenčnej oblasti na pozícii (u,v) v `I` rámci. Nech $\Delta F(u,v)$ značí zmenu hodnoty DCT koeficientu. Potom zmena variancie hodnôt bodov v celom `I` rámci je daná nasledovne:

$$\begin{aligned}
\Delta D(f(x,y)) &= E(f(x,y)+\Delta f(x,y))^2 - E^2(f(x,y)+\Delta f(x,y)) \\
&- Ef^2(x,y) + E^2(f(x,y)) \\
&= E((f(x,y)+\Delta f(x,y))^2 - E(f^2(x,y))) \\
&- E^2(f(x,y)+\Delta f(x,y)) + E^2(\Delta f(x,y)) \\
&= E(\Delta f(x,y))^2 - (E\Delta f(x,y))^2 \\
&= \frac{4}{MN}c^2(u)c^2(v)[E(\Delta F(u,v).\cos\frac{(2x+1)u\pi}{2M}.\cos\frac{(2y+1)v\pi}{2N})^2 \\
&- (E(\Delta F(u,v).\cos\frac{(2x+1)u\pi}{2M}.cos\frac{(2y+1)v\pi}{2N}))^2], \qquad (3.1)
\end{aligned}$$

kde pre $x=0$ je $c(x)=1/\sqrt{2}$ a pre $x=1,2,\ldots,M-1$, resp. pre $x=1,2,\ldots,N-1$ je $c(x)=1$. M a N označujú veľkosť obrazového rámca.

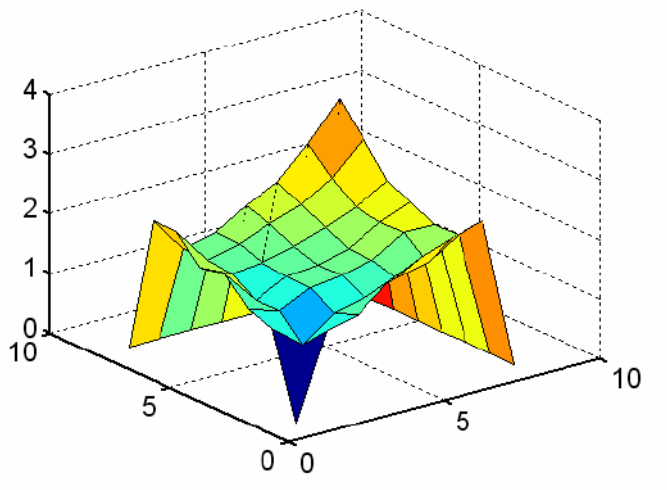
Ak je jeden vybraný DCT koeficient zmenený o 1, zmenu variancie v priestorovej oblasti možno určiť použitím vzťahu 3.1, ktorý v sebe zahŕňa transformáciu sledovaného DCT koeficientu z frekvenčnej do priestorovej oblasti pomocou inverznej diskrétnej kosínusovej transformácie.

Násobením s intrakódovanou kvantizačnou maticou možno určiť efekt variancie pre každý kvantizovaný DCT koeficient, viď tabuľka 3.2.

Obrázok 3.5 predstavuje trojrozmernú reprezentáciu dát z tabuľky 3.2. Na obrázku vidno, ako vplyv na varianciu možno rozdeliť do štyroch rôznych skupín.

Tabuľka 3.2: Vplyv na varianciu jednotlivých kvantizovaných DCT koeficientov

v / u	1	2	3	4	5	6	7	8
1	0	1.4142	1.5343	1.9445	2.2094	2.3865	2.5511	0.0000
2	1.4142	1.0000	1.3750	1.5000	1.6875	1.8125	2.1250	0.0000
3	1.5343	1.3750	1.6129	1.6875	1.8065	2.1250	2.1241	0.0000
4	1.9445	1.3750	1.6250	1.6875	1.8125	2.1250	2.3125	0.0000
5	1.8695	1.6250	1.6819	1.8125	1.9970	2.1875	2.4995	0.0000
6	2.2981	1.6875	1.8125	2.0000	2.1875	2.5000	3.0000	0.0000
7	2.2872	1.6875	1.8118	2.1250	2.3746	2.8750	3.4999	0.0000
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000



Obr. 3.5: Vplyv na varianciu jednotlivých kvantizovaných DCT koeficientov [51].

Algoritmus vkladania v komprimovanej oblasti

V tomto algoritme kladné párne a záporné nepárne DCT koeficienty reprezentujú nulu, kladné nepárne a záporné párne koeficienty reprezentujú jednotku. Ďalej, hodnota 1 má byť modifikovaná na -1 alebo 2, a hodnota -1 má byť modifikovaná na -2 alebo 1.

Vstup: video nosič U, šifrovací kľúč key, správa M.

Výstup: stego-video X, koeficient prispôsobenia β.

Predspracovanie:

Krok 1: Zašifrovanie správy: $C \leftarrow E_{key}(M)$.

Krok 2: Pre každý I rámec F_i sa vypočíta hodnota variancie jednotlivých bodov $\sigma^2_{F_i}$ s DCT DC koeficientami.

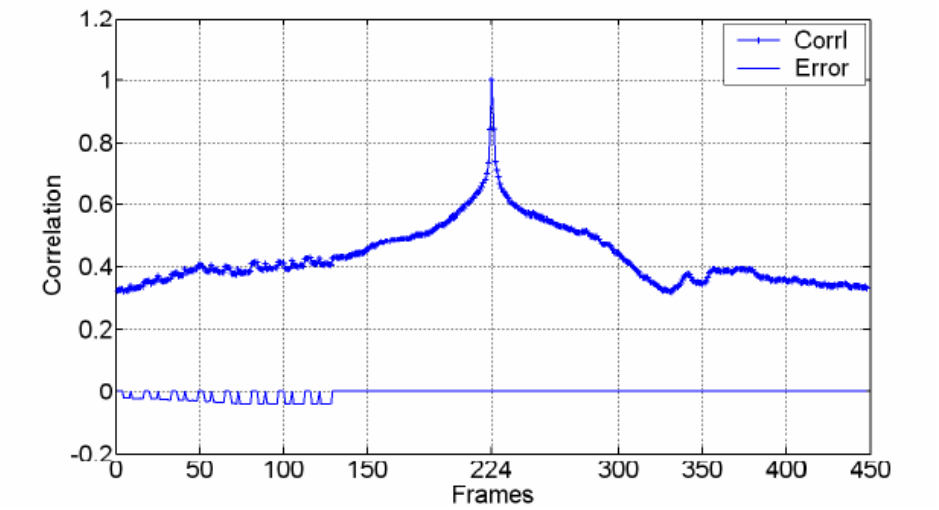
Krok 3: Podľa hodnoty $\sigma^2_{F_i}$ sa pridelí tzv. zaťaženie pre každý rámec

$$s_i \leftarrow \frac{|S|\sigma^2_{F_i}}{\beta \sum \sigma^2_{F_i}}.$$

Krok 4: Pre každé s_i sa priradí bit správy k dostupným DCT AC koeficientom a odhadne sa hodnota ΔD.

Krok 5: Vypočíta sa korelácia medzi $\sigma^2_{F_i}$ a ΔD. Ak $Corr(\sigma^2_{F_i}, \Delta D) < T$, tak choď na Krok 3 a prispôsob koeficient β.

Krok 6 (vloženie): Pre každé s_i, sa spracuje bit správy, ktorý sa vkladá pomocou DCT koeficientov.



Obr. 3.6: Zmena v korelácii medzi snímkami pri sekvenčnom vkladaní v úvodnej časti videa [51].

V navrhovanom algoritme sa správa rozdelí do N častí a vkladá do `I` rámcov. V prípade sekvenčného vkladania môže dôjsť k zmenám v korelácii medzi snímkami, ako je to možné vidieť na obrázku 3.6. Na tomto obrázku je znázornená korelácia rámca č. 224 so všetkými ostatnými rámcami v MPEG–2 videu `Mobl_080.m2v`. Sekvenčné vkladanie bolo aplikované v prvých rámcoch až po rámec č. 126, pričom v tejto oblasti možno pozorovať výrazné narušenie hladkosti korelačnej krivky. Táto skutočnosť by určite mohla byť využitá pre stegoanalýzu.

3.5 Modifikácia pohybových vektorov

V tejto časti uvádzaný algoritmus (spracované podľa [83]) sa zaoberá technikou steganografie, ktorá pracuje s komprimovaným video prúdom. Preto nie je potrebné zaoberať sa konkrétnym kódekom, ktorý vytváral video prúd. Tajné údaje sú vkladané do tzv. *pohybových vektorov* zmien jednotlivých makroblokov, definovaných v `P` alebo `B` snímkach. Aby sa predišlo strate informácie počas prenosu (výpadkom niektorých snímok v zašumenom kanále), informácia sa vkladá opakovane (vnáša sa dodatočná redundancia). Na uľahčenie dekódovania sa niektoré riadiace informácie ukladajú priamo v `I` snímkach.

Steganografický algoritmus

Video prúd v štandarde MPEG je zložený z hlavičky, DCT údajov a postupnosti pohybových vektorov. DCT údaje sú tvorené procesom kódovania `I` snímok, postupnosť pohybových vektorov zodpovedá postupnosti `P` alebo `B` snímok. Tieto sú použité na kompenzáciu predikcie pohybu jednotlivých makroblokov, z ktorých sa každý obrázok skladá. V `P` type obrázkoch každému makrobloku zodpovedá jeden pohybový vektor. V `B` snímkach má každý makroblok dva pohybové vektory, pretože výsledná predikcia sa určuje pomerným priemerom oboch vektorov, pričom jeden z nich zodpovedá predošlému, a druhý nasledujúcemu referenčnému snímku. Ako už bolo viackrát spomenuté v predošlom texte, referenčnými snímkami môžu byť buď `P` alebo `I` snímky. Vo výslednom video prúde sa nachádza oveľa viac `P` a `B` snímok než obrázkov typu `I`. Preto je vhodné hľadať také techniky vkladania informácie, ktoré zohľadňujú túto štruktúru štandardu, a vkladajú informáciu spôsobom, ktorý čo najmenej modifikuje cieľový údajový prúd. Ukazuje sa, že práve pohybové vektory sú vhodné na takéto využitie. Nachádzajú sa v každom `P` a `B` type obrázka, a nakoľko sa jedná o predikciu, je samotnou podstatou stanovená tolerancia istých odchýlok od originálneho obrazu. Táto odchýlka môže byť využitá aj pre kódovanie tajnej správy.

V závislosti od typu videa je možné uvažovať o rôznych transformáciách, ktoré môžu byť s videom robené za účelom zefektívnenia prenosu: ak sú obrázky v istej časovej postupnosti približne rovnaké, môže sa stať, že niektoré z nich sa zahodia, alebo sa poprehadzuje ich poradie. Aby bolo možné tajnú informáciu neskôr zrekonštruovať, je nutné zaviesť istú mieru redundancie, ktorá zvýši odolnosť prežitia správy v prenosovom kanále.

Algoritmus vkladania tajnej informácie nevyužíva na vkladanie každý pohybový vektor. Dôvodom je to, že v prípade zmeny každého pohybového vektora by sa zaviedol pozorovateľný šum do výsledného obrazového prúdu. Napríklad, pokiaľ by sa viaceré obrázky po sebe významne nelíšili, zakódovaním tajnej informácie by sa vložili pozorovateľné a identifikovateľné zmeny medzi týmito snímkami. Preto je nutné, podobne ako pri vkladaní tajnej informácie v priestorovej oblasti do LSB bitov farebných zložiek obrazu, sledovať isté charakteristiky monotónnosti cieľovej skupiny obrazov idúcich po sebe. Pokiaľ sa líšia iba málo, informácia sa nesmie vložiť. To, či je zmena dostatočne veľká, určuje voliteľný parameter algoritmu. Podrobnosti sú uvedené ďalej. Celá metóda modifikácie pohybových vektorov je odvodená od algoritmu autorov článku [86].

Nasleduje opis algoritmu:

Krok 1: Urob zoznam všetkých pohybových vektorov (PVM) v danej `P` alebo `B` snímke.

Krok 2: Pre všetky pohybové vektory v zozname vypočítaj absolútnu veľkosť podľa vzťahu

$$|PVM_i| = \sqrt{H_i^2 + V_i^2},$$

kde H_i je horizontálna a V_i je vertikálna zložka pohybového vektora.

Krok 3: Zo zoznamu vyber iba také, ktorých absolútna veľkosť je väčšia než špecifikovaná tolerancia ϵ. Podľa tejto hranice vyber makrobloky, ktoré je možné použiť na vloženie informácie:

$$MB_i = \begin{cases} 1 & |PVM_i| > \epsilon \\ 0 & |PVM_i| < \epsilon \end{cases},$$

kde $MB_i = 1$ označuje makroblok, ktorý spĺňa kritérium na vloženie informácie, kým $MB_i = 0$ je označenie bloku nevyhovujúceho kritériu vkladania.

Krok 4: Tento výpočet urob pre všetky `P` a `B` snímky skupiny obrázkov (GOP). Tým sa získa počet použiteľných pohybových vektorov. Do každého takéhoto vektoru je možné vložiť jeden bit informácie. Zistením počtu použiteľných makroblokov v celom videu sa určí kapacita média. Kapacitu je ešte potrebné upraviť podľa redundancie (príslušne znížiť). V tejto verzii algoritmu sa na redundanciu používa štvornásobné kopírovanie informácie (takže celková kapacita sa musí vydeliť štyrmi) — GOP sa rozdelí na 4 časti, a do každej sa uložia rovnaké informácie. Pri extrakcii sa pomocou majoritnej zhody určí, aká informácia bola vložená.

Krok 5: Nasleduje samotné vloženie údajov do určených makroblokov. Najprv sa vypočíta uhol pohybového vektora:

$$\theta_i = \arctan \frac{V_i}{H_i}.$$

Na základe jeho veľkosti sa uskutoční zmena podľa tejto schémy:

1. Ak je uhol ostrý, menšiu chybu spôsobí modifikácia horizontálnej zložky pohybového vektora, preto sa bit informácie vloží tam:
   - Ak $\mathrm{mod}(2 * H_i, 2) = M_k$, potom H_i sa nemení,
   - ale ak $\mathrm{mod}(2 * H_i, 2) \neq M_k$, potom $H_i = H_i + 0.5$.

   H_i je horizontálna zložka pohybového vektora i, `mod` je funkcia modulárneho zvyšku, M_k je k–ty bit tajnej informácie.
2. Ak je uhol tupý, menšia chyba bude spôsobená upravením vertikálnej zložky pohybového vektora. Preto sa modifikácia uskutoční nasledovne:
   - Ak $\mathrm{mod}(2 * V_i, 2) = M_k$, potom V_i sa nemení,
   - ale ak $\mathrm{mod}(2 * V_i, 2) \neq M_k$, potom $V_i = V_i + 0.5$.

   V_i je vertikálna zložka pohybového vektora i, `mod` je funkcia modulárneho zvyšku, M_k je k–ty bit tajnej informácie.

Krok 6: V skupine obrazov (GOP) sa týmto spôsobom do každého zo štyroch definovaných segmentov (vnesená redundancia informácie) vloží tajná informácia. Podobne sa vyplnia všetky skupiny obrazov v snímke a všetky snímky vo videu.

Vkladanie kontrolnej informácie

`I` snímka je prvou snímkou v skupine obrazov (GOP). Informácie, ktoré sa ukrývajú do `I` snímok, uľahčujú extrakciu údajov. Nakoľko sa v algoritme využíva vkladanie redundantnej informácie, je potrebné vložiť isté riadiace informácie ohľadom umiestnenia a rozdelenia štyroch segmentov určených pre tú istú informáciu. Okrem definovania oblastí sa ukladá informácia o kapacite média — tým sa uľahčuje kontrola správnosti dekódovania. Takže začiatočná snímka skupiny obrazov obsahuje:

- Kapacitu média a
- začiatočnú a koncovú pozíciu každého zo štyroch redundačných segmentov v GOP–e.

Tieto informácie sa na rozdiel od `P` a `B` snímok ukladajú do DCT koeficientov jasovej a (alebo) farebnej obrazovej zložky. Informácia sa ukladá do stredných frekvencií kvantizovaných DCT koeficientov, nakoľko vloženie do nízkych by znížilo robustnosť aplikácie (modifikácia zmenou šumu v prenosovom kanáli alebo prípadnou manipuláciou s grafickým objektom), a vloženie do vysokých frekvenčných zložiek by spôsobilo viditeľný šum dekódovaného obrázka. Takto sa dosahuje správny pomer (angl. *tradeoff*) medzi robustnosťou a nenápadnosťou steganografického média.

Pri procese vkladania sa najprv dekóduje sled DCT koeficientov, ktoré sa použijú na vloženia riadiacich informácií. Vkladanie sa realizuje pomocou techniky modifikácie najmenej významného bitu zvolených DCT koeficientov (LSB technika). Podrobnejšie túto techniku opisuje článok [80]. Po vložení sa DCT koeficienty opäť zakódujú pomocou VLC a uložia do výstupného bitového prúdu.

Algoritmus extrakcie

Je zrejmé, že najprv je potrebné dekódovať riadiace informácie z prvej snímky každej skupiny obrazov. Dekódovaním DCT koeficientov a extrahovaním riadiacej informácie sa zistí kapacita média (alebo dĺžka vloženej správy, v závislosti od implementácie) a počiatočné a koncové indexy segmentov, ktoré slúžia na zvýšenie robustnosti systému. Samotné získavanie tajnej informácie z `P` a `B` snímok je nasledovné:

Krok 1: Urob zoznam všetkých pohybových vektorov (PVM) v danej `P` alebo `B` snímke.

Krok 2: Pre všetky pohybové vektory v zozname vypočítaj absolútnu veľkosť podľa vzťahu

$$|PVM_i| = \sqrt{H_i^2 + V_i^2},$$

kde H_i je horizontálna a V_i je vertikálna zložka pohybového vektora.

Krok 3: Zo zoznamu vyber iba také, ktorých absolútna veľkosť je väčšia než špecifikovaná tolerancia ϵ. Podľa tejto hranice vyber makrobloky, ktoré je možné použiť na vloženie informácie:

$$MB_i = \begin{cases} 1 & |PVM_i| > \epsilon \\ 0 & |PVM_i| < \epsilon \end{cases},$$

kde $MB_i = 1$ označuje makroblok, ktorý spĺňa kritérium na vloženie informácie, kým $MB_i = 0$ je označenie bloku nevyhovujúceho kritériu vkladania.

Krok 4: Nasleduje samotné extrahovanie údajov z určených makroblokov. Najprv sa vypočíta uhol pohybového vektora:

$$\theta_i = \arctan \frac{V_i}{H_i}.$$

Na základe jeho veľkosti sa uskutoční extrakcia podľa tejto schémy:

1. Ak je uhol ostrý, údaje boli vložené do horizontálnej zložky a získajú sa pomocou vzťahu:
$$M_k = \mathrm{mod}(2 * H_i, 2),$$
kde H_i je horizontálna zložka pohybového vektora i, `mod` je funkcia modulárneho zvyšku, M_k je k–ty bit tajnej informácie.
2. Ak je uhol tupý, údaje boli vložené do vertikálnej zložky a získajú sa pomocou vzťahu:
$$M_k = \mathrm{mod}(2 * V_i, 2),$$
kde V_i je horizontálna zložka pohybového vektora i, `mod` je funkcia modulárneho zvyšku, M_k je k–ty bit tajnej informácie.

Krok 5: Táto extrakcia sa urobí pre všetky `P` a `B` snímky v GOP–e. Keďže sa pri prenose predpokladá modifikácia časti informácie v dôsledku zašumenia prenosového kanálu, je potrebné určiť prenášanú tajnú správu na základe redundantných informácií. Toto určenie je stanovené pomocou majoritnej zhody bitov zo štyroch segmentov, ktoré by mali obsahovať tú istú informáciu.

Krok 6: Získanie tajnej informácie pokračuje pre všetky skupiny obrázkov vo videu.

Kapitola 4

Steganografický systém založený na súborovom formáte BMP

Na začiatku tejto časti objasníme základnú štruktúru obrazového formátu BMP. Potom uvedieme návrh steganografického systému založenom na predmetnom súborovom formáte. Napokon venujeme pozornosť stegoanalýze navrhnutého systému.

4.1 Štruktúra súborového formátu BMP

Bitmapové obrázky, medzi ktoré patria všetky reprezentované súborovým formátom BMP, uchovávajú informácie o farbách každého obrazového bodu. Počet možných farieb úzko súvisí s počtom bitov vyhradených na uloženie farieb. V časti 2.1 sme o tom trošku hovorili. V súčasnosti sa používa ukladanie troch farebných zložiek (alebo alternatívne ukladanie jasových zložiek v transformovanej oblasti), pričom pre každú farbu je vyhradené pamäťové miesto veľkosti jedného bajtu (t.j. 8 bitov). To znamená, že jednému obrazovému bodu zodpovedá $8 * 3 = 24$ bitov. V minulosti sa používali farebné modely s bitovou hĺbkou 1, 4, 8 alebo 16 bitov. V prípade, že okrem informácií o jednotlivých farebných (resp. jasových) zložkách sa uchováva aj informácia o priehľadnosti obrazového bodu, k uvedeným 24 bitom sa pridáva ďalších 8 bitov uchovávajúcich túto informáciu.

Bitmapové obrázky sa skladajú z hlavičky, bitmapových údajov a prípadne ďalších informácií. Hlavička pozostáva z údajov, ktoré obsahujú informácie o bitmape umiestnenej ďalej v súbore.

Súborový formát Microsoft Windows Bitmap (BMP) je jedným z viacerých formátov, ktoré reprezentujú grafickú informáciu pomocou bitmapy. Hoci je BMP formát dobre definovaný, nejestvujú oficiálne špecifikácie publikované firmou, ktorá ho vyvinula (Microsoft). Informácie o tejto štruktúre sa nachádzajú v príručkách, manuáloch a iných súboroch balíka Microsoft Windows Software Development Kit (SDK).

Súbor obsahuje štyri sekcie: bitmapovú hlavičku, informačnú hlavičku, paletu a bitmapu. Bitmapová hlavička je 14 bajtov dlhá, informačná hlavička 40 bajtov. Paleta a bitmapa majú premenlivú dĺžku.

Hlavička súboru má nasledovnú štruktúru:

```
Bitmapova Hlavicka:
    TypSuboru           velkost 2B
    VelkostSuboru       velkost 4B
    Rezervovane         velkost 2B
    Rezervovane         velkost 2B
    PosunBitmapy        velkost 4B
```

`TypSuboru` obsahuje číslo identifikujúce typ súboru. Vždy je jeho hexadecimálna hodnota 0x4D42h. `VelkostSuboru` uchováva celkovú veľkosť súboru v bajtoch. Táto hodnota by sa mala zhodovať s veľkosťou udávanou operačným systémom. `Rezervovane` polia majú vždy hodnotu 0 a nepoužívajú sa. `PosunBitmapy` určuje začiatok bitmapy ako posunutie od začiatku súboru k začiatku obrazových údajov. Posunutie sa určuje v bajtoch.

Informačná hlavička obsahuje nasledovné informácie o uchovávaných obrazových údajoch:

```
Informacna Hlavicka:
    VelkostHlavicky     velkost 2B
    SirkaObrazku        velkost 4B
    VyskaObrazku        velkost 4B
    PocetRovin          velkost 1B
    BityNaPixel         velkost 1B
    KompresnaMetoda     velkost 2B
    VelkostBitmapy      velkost 2B
    HorizontRozlisenie  velkost 2B
    VertikalRozlisenie  velkost 2B
    PocetFarieb         velkost 2B
    PocetDolezitychF    velkost 2B
```

`VelkostHlavicky` udáva veľkosť informačnej hlavičky v bajtoch. Táto hodnota sa zároveň využíva na identifikáciu verzie bitmapového typu súboru. `SirkaObrazu` a `VyskaObrazu` označujú šírku a výšku obrazu v obrazových bodoch. `PocetRovin` sa vždy rovná jednej, pretože obrázky `BMP` obsahujú vždy iba jedinú farebnú rovinu. `BityNaPixel` môžu nadobúdať hodnotu 1, 4, 8, 16, 24 alebo 32. `KompresnaMetoda` indikuje typ kódovania použitého na bitmapu. Hodnota 0 znamená, že údaje nie sú komprimované. 1 indikuje osembitovú `RLE` kompresiu a 2 štvorbitovú `RLE` kompresiu. `VelkostBitmapy` je veľkosť komprimovaných údajov bitmapy uvádzaná v bajtoch. Ak nie je použitá komprimácia, táto veličina má hodnotu 0. Vtedy program počíta veľkosť bitmapy z rozmerov obrázka a použitej bitovej hĺbky. `HorizontRozlisenie` a `VertikalRozlisenie` uvádzajú horizontálne a vertikálne hodnoty rozlíšenia v obrazových bodoch na meter. Používajú sa pre voľbu vhodného režimu pre tlač alebo zobrazenie obrázka. `PocetFarieb` obsahuje počet farieb použitých v palete. Ak je táto položka nulová, počet použitých farieb sa rovná maximálne možnému počtu pre danú bitovú hĺbku. Táto hodnota sa potom vypočíta vzťahom `PocetFarieb = 1 << BityNaPixel`, kde operátor `<<` znamená bitový posun ľavého operandu vľavo o hodnotu pravého operandu. `PocetDolezitychFarieb` určuje počet významných farieb v palete. Tento počet sa zisťuje podľa ich frekvencie výskytu v obrázku: čím častejšie sa nejaká farba v obrázku vyskytuje, tým viac narastá jej významnosť. Toto pole bolo definované kvôli hardvéru, ktorý podporoval menej farieb, než obrázok používa. Ak sú všetky farby v bitmape významné, toto pole nadobúda hodnotu 0.

Všetky jedno, štvor a osembitové obrázky `BMP` majú paletu. Veľkosť palety závisí na počte farieb v obrázku a je určená hodnotou v poli `PocetFarieb` v informačnej bitmapovej hlavičke.

Zvyšok `BMP` súboru pozostáva z vlastných bitmapových údajov, ktoré môžu byť buď nekomprimované, alebo uložené metódou `RLE`. Obrazové údaje v `BMP` súbore sú vždy bajtovo orientované. Ich interpretácia však závisí od použitej bitovej hĺbky a od toho, či sú údaje komprimované alebo nie.

V jednobitových obrázkoch reprezentuje každý bajt osem obrazových bodov, pričom najvýznamnejší bit v bajte je prvá pixelová hodnota. Štvorbitové obrázky sú uložené ako dva obrazové body na jeden bajt, pričom významnejšia štvorica je prvým bodom a menej významná štvorica druhým obrazovým bodom. Osembitové obrázky obsahujú jeden obrazový bod na bajt. 16 bitové obrázky ukladajú informácie o jednotlivých farebných zložkách do troch pätíc bitov. Niekedy sa pre zelenú zložku využíva miesto piatich bitov šesť, pretože ľudské oko je najcitlivejšie na zelenú zložku farebného spektra. Zväčšením bitového rozsahu sa rozšíri zoznam použiteľných farieb pre túto farebnú zložku. 24 bitové obrázky využívajú 3 bajty na uloženie farebných informácií o jednom obrazovom bode. Farby sa ukladajú v poradí červená, zelená a modrá zložka. V prípade použitia 24 bitovej hĺbky ostáva posledný bajt (*alfa kanál*) nevyužitý. Ten sa používa pri obrázkoch s 32 bitovou farebnou hĺbkou na uloženie informácie o priehľadnosti obrazového bodu.

Každý riadok bitmapy v BMP súbore musí byť násobkom štyroch. Ak to je potrebné, chýbajúce bajty sa doplnia nulami. Obrazové údaje sú vždy ukladané do bitmapy od ľavého dolného rohu obrazovky, teda obrazový bod (0,0) je ľavým dolným rohom obrázka.

4.2 Návrh steganografického systému

Navrhli sme implementovať dva základné moduly systému: modul pre prácu so steganografickým médiom (v našom prípade s obrázkami rôznych grafických formátov) a modul pre riadenie procesu vkladania správy (jedná sa o modul, ktorý implementuje rôzne steganografické algoritmy vkladania správ do nosičov).

V rámci implementácie aplikačného prototypu sme venovali pozornosť spracovaniu bitmapových grafických súborov `BMP`. Väčšina komerčných steganografických a stegoanalytických systémov umožňuje prácu s týmto súborovým formátom, preto pre plánovanú stegoanalýzu bolo jednoduchšie urobiť výstupné analýzy na tomto súborovom formáte. Okrem toho sa jedná o reprezentáciu obrazu, ktorá sa používa v nekomprimovanom videu. Na vkladanie správy sme navrhli jednoduchý sekvenčný algoritmus LSB.

V navrhovanom systéme vystupujú nasledovné entity:

- steganografický nosič,
- steganografický algoritmus,
- správa.

Správa a výber steganografického algoritmu sú parametrami funkcie vkladania informácie do steganografického média. Pri funkcii extrahovania je vstupným parametrom steganografický nosič, z ktorého sa získa informácia o použitom algoritme a vloženej správe.

V procese návrhu steganografického systému sme navrhli niekoľko testov, ktoré overili úroveň zhody implementovaného prototypu so špecifikáciou. Jednalo sa o nasledovnú sadu testov:

1. vloženie správy algoritmom LSB do jedného obrázka BMP,
2. vloženie správy algoritmom LSB do viacerých obrázkov BMP,
3. nesprávna špecifikácia vstupu v móde vkladania správ,
4. extrahovanie správy zo steganografického obrázka BMP,
5. extrahovanie správy z viacerých obrázkov BMP,
6. nesprávna špecifikácia vstupu v móde extrahovania správ a
7. nosič v móde extrahovania správ neobsahuje steganografickú informáciu uloženú LSB metódou.

4.3 Stegoanalýza LSB

Základné typy útokov na algoritmus LSB sa dajú rozdeliť do nasledujúcich troch kategórií:

Vizuálne útoky — Útoky z tejto skupiny filtrujú významné zložky obrázka tak, aby človek mohol vizuálne identifikovať anomálie. Základný test z tejto skupiny zobrazuje obrázok podľa najmenej významného bitu (tzv. *LSB* bit), v ktorom náhodný šum často odkrýva existenciu ukrytej správy. Je to z dôvodu, že nedokonalé kamery, skenery a ďalšie prístroje zanechávajú spravidla odlesky veľkých štruktúr v LSB bitoch.

Štrukturálne útoky — Útoky z tejto skupiny využívajú fakt, že pôvodné údaje môžu po ukrytí správy dramaticky zmeniť niektoré vlastnosti. Často sa dajú tieto zmeny ľahko identifikovať v samotnej štruktúre údajov.

V niektorých prípadoch Napríklad používajú steganografické algoritmy jemne odlišné verzie formátu súboru. V iných prípadoch zasa môžu dopĺňať dodatočné informácie spôsobom, ktorý sa odlišuje od spôsobu manipulácie iných aplikácií.

Štatistické útoky — Charakter jednotlivých bodov a ich najmenej významných bitov často umožňuje odhaliť existenciu ukrytej správy pomocou štatistických metód. Pri týchto typoch útokov sa využíva skutočnosť, že vkladané údaje majú zväčša iný štatistický profil ako pôvodné dáta.

V ďalšom sa budeme zaoberať len štatistickým testom, keďže štrukturálne útoky v prípade vkladania informácií do LSB bitov obrázkov vo formáte BMP možno vylúčiť a úspešnosť vizuálneho útoku je spravidla veľmi nízka. Navyše úspešnosť vizuálneho útoku sa dá vyvodiť z úspešnosti štatistického útoku, a preto tento považujeme za najsilnejší spomedzi spomenutých útokov.

Poznámka 4. *V ďalšom budeme označovať úspešnosť štatistického útoku na oblasť O symbolom $P(O)$. Jej hodnoty budeme vyjadrovať v percentách. To znamená, že oblasť O je ideálna pre vloženie správy, ak $P(O) = 0$ a je úplne nevhodná pre $P(O) = 1$.*

4.4 Štatistický útok a χ^2 test

χ^2 test predstavuje základnú metódu, ktorá sa používa na analýzu vlastností náhodných postupností. Často sa pomocou neho vyhodnocuje miera odchýlky pozorovaných veličín od teoreticky očakávaných. Vo všeobecnosti sa dá popísať tento test nasledovne.

Prepokladajme, že uskutočníme n krát experiment, ktorého výsledok môže nadobudnúť k rôznych hodnôt $\{h_1, \ldots, h_k\}$.

Poznámka 5. *χ^2 test, kde skúmame výskyt k rôznych hodnôt, voláme χ^2 test s $k-1$ stupňami voľnosti a označujeme ho symbolom χ^2_{k-1}.*

Nech symbol Y_i označuje počet výskytov hodnoty h_i medzi realizovanými náhodnými pozorovaniami, pričom výsledok jedného pozorovania nemá vplyv na žiadne ďalšie. Ďalej nech p_i označuje očakávanú hodnotu pravdepodobnosti výskytu h_i. Hodnota

$$\chi^2 = \sum_{i=1}^{k} \frac{(Y_i - np_i)^2}{np_i} \tag{4.1}$$

označuje mieru, v akej sa líši pozorovaná veličina od očakávanej. Otázkou zostáva, ako interpretovať túto hodnotu z hľadiska „podobnosti“ oboch veličín. Jedná sa o to, ako interpretovať chybu H_0 prvého druhu, t.j. aká je pravdepodobnosť, že pozorovaná veličina s hodnotou χ^2 má rovnaké rozdelenie pravdepodobnosti ako veličina očakávaná. Toto sa zakladá na χ^2 rozdelení náhodnej premennej, keďže veličina χ^2 má v limitnom prípade charakter χ^2 rozdelenia s $k-1$ stupňami voľnosti. Od toho sa odvíja aj spôsob interpretácie konkrétnej hodnoty χ^2 testu.

Pri útoku χ^2 testom predpokladáme dva možné prístupy. V jednom z nich máme k dispozícii dáta, ktoré chceme vložiť a v druhom nie.

Poznámka 6. *V druhom prípade sa domnievame, že príslušné údaje, ktoré sa neskôr do obrazu vložia, majú charakter náhodnej postupnosti.*

Pre jednoduchosť predpokladáme nasledovné skutočnosti vkladania dát:

- informácie sú vkladané do najmenej významných bitov oblasti O a
- oblasť O, do ktorej vkladáme, je obdĺžnikového tvaru a je určená dvoma bodmi P_{min} a P_{max} so súradnicami $P_{min} = [x_{min}, y_{min}]$ a $P_{max} = [x_{max}, y_{max}]$.

Naším cieľom je určiť vhodnosť vybranej oblasti pre steganografické účely. Snažíme sa určiť, do akej miery sa zmení štatistický profil oblasti po vložení správy. Na tento účel používame χ^2 test, pomocou ktorého budeme zisťovať štatistické odchýlky oblastí pred a po vložení údajov.

Najskôr vytvoríme reťazec R LSB bitov oblasti O tak, že postupne extrahujeme LSB bity bodov so súradnicami $[x_{min}, y_{min}], [x_{min}+1, y_{min}], \ldots, [x_{max}, y_{min}], [x_{min}+1, y_{min}+1], \ldots$. Dostávame tak reťazec R dĺžky

$$d = (x_{max} - x_{min})(y_{max} - y_{min}).$$

V ňom zistíme počet výskytov jednotlivých reťazcov r_i dĺžky l bitov. V súlade s označením z popisu χ^2 testu budeme početnosť výskytu reťazca r_i v reťazci R označovať symbolom Y_i.

Obr. 4.1: Pôvodný obrázok

Následne realizujeme χ^2 test s $2^l - 1$ stupňami voľnosti. Ako hodnoty p_i do vzorca 4.1 berieme za prepokladanú pravdepodobnosť p_i výskytu reťazca r_i v R hodnotu $p_i = 2^{-l}$ (to v prípade, keď nepoznáme vkladané údaje).

Hodnota χ^2 testu má teda nasledovný tvar:

$$\chi^2 = \sum_{i=1}^{2^l} \frac{(Y_i - d2^{-l})^2}{d2^{-l}} \tag{4.2}$$

V prípade, kde poznáme vkladané údaje (reťazec R' dĺžky d), vezmeme za p_i relatívnu početnosť výskytu reťazca r_i. Ak Y_i' označuje početnosť výskytu r_i v R', potom má χ^2 tvar:

$$\chi^2 = \sum_{i=1}^{2^l} \frac{(Y_i - Y_i')^2}{Y_i'} \tag{4.3}$$

Úspešnosť štatistického útoku budeme brať ako hodnotu pravdepodobnosti, s akou nadobúda pôvodný obrázok a stego–správa rovnaké rozdelenie reťazcov r_i pre $i \in \{1, \ldots, 2^l - 1\}$.

Obr. 4.2: Filtrovaný obrázok

4.5 Experimenty

V experimentoch sme analyzovali obrázok 4.1. Jeho filtrovaný obraz predstavuje obrázok 4.2. Biele body predstavujú „jednotkové“ LSB bity originálu, čierne body zasa „nulové“ bity.

Programom sme analyzovali úspešnosť štatistického testu. Zisťovali sme početnosť výskytu všetkých reťazcov dĺžky $l \in \{1,2,\ldots,8\}$. Pomocou týchto početností sme zistili nasledovné hodnoty χ^2 testu s $n = 2^l - 1$ stupňami voľnosti, ktoré uvádzame v tabuľke 4.1.

n	1	3	7	15	31	63	127	255
χ^2 R	4952	141119	346428	670370	1.1×10^6	1.94×10^6	3.14×10^6	5.04×10^6
χ^2 G	692	55521	127051	226311	358486	534532	770055	1.08×10^6
χ^2 B	51	10957	23518	38467	55422	74930	97281	123442

Tabuľka 4.1: Hodnoty χ^2 testu pre zložky R, G a B.

V ďalšom kroku sme vložili do obrázka zašifrovanú správu. Získali sme nasledujúce hodnoty χ^2 testov, ktoré uvádzame v tabuľke 4.2 spolu s úspešnosťou štatistického testu (hodnoty $1 - F_n$).

Následne sme sa zamerali na menšiu oblasť obrázka. Konkrétne sa jednalo o oblasť určenú bodmi so súradnicami $[0,0], [100,100]$. Výsledky sumarizuje tabuľka č. 4.3.

n	1	3	7	15	31	63	127	255
χ^2	0.06116	5.919	14.04	30.66	48.31	84.06	149.37	275.56
$1-F_n(\chi^2)$	0.804672	0.115618	0.0504745	0.00975195	0.02457	0.039353	0.0853398	0.179737

Tabuľka 4.2: Hodnoty χ^2 testu po vložení správy.

n	1	3	7	15	31	63	127	255
χ^2 R	653.314	4773.48	11574.8	23271.1	42656.3	74573.6	128841	220326
χ^2 G	291.044	2172.5	4725.5	8438.36	13692.6	20955.7	31376.5	45981.1
χ^2 B	0.81	235.098	498.232	809.818	1171.67	1569.72	2038.13	2667.64

Tabuľka 4.3: Hodnoty χ^2 testu pre R, G a B zložky vybranej oblasti obrázka.

Potom sme vložili do oblasti zašifrovanú správu. Získali sme nasledujúce hodnoty χ^2 testov, ktoré uvádzame v tabuľke 4.4 a 4.5 spolu s úspešnosťou štatistického testu (hodnoty $1-F_n(\chi^2)$).

n	1	3	7	15	31	63	127	255
χ^2	0.36	0.7776	4.7552	13.5424	28.0897	61.1667	123.923	253.077
$1-F_n(\chi^2)$	0.548506	0.854817	0.689808	0.560479	0.616531	0.541953	0.56069	0.522258

Tabuľka 4.4: Hodnoty χ^2 testu po vložení správy do oblasti 100×100.

n	1	3	7	15	31	63	127	255
χ^2 R	7.6	29.6	29	28	26	26	24	23
χ^2 G	2.37	25.6	27	27	26.3	25.5	24.5	23.5
χ^2 B	63	46	47	47.5	47.3	47.7	47.7	46.2

Tabuľka 4.5: Podiely hodnôt χ^2 testu obrázka ku oblasti 100×100 pre R, G a B zložky.

Nakoniec sme zisťovali hodnoty χ^2 testu pre oblasti určené bodmi $P_{min} = [i.100, j.100]$ a $P_{max} = [(i+1).100, (j+1).100]$ (tabuľka 4.6).

i/j	0	1	2	3	4	5	6	7
0	2667.64	958.411	1468.36	5968.56	4879.18	701.282	2232.36	2797.08
1	3161.85	1114.67	1382.36	6440.26	8999.02	812.154	4838.41	3886.92
2	2856.21	995.949	1895.9	5594.41	5132.56	816.052	1707.69	3071.74
3	2624	1142.82	1887.9	5850.15	9670.66	789.692	4310.05	4668.1
4	2478.31	1588.56	2316.72	5456.77	4998.97	933.693	2499.44	2726.05
5	3410	895.385	1459.44	6607.02	11175.7	587.385	3878.36	4139.69

Tabuľka 4.6: Hodnoty χ^2 testu pre B zložku obrázka a jednotlivé oblasti velkosti 100×100.

4.6 Vyhodnotenie experimentov

- Správa, ktorá bola vkladaná do obrázka, nemala charakter náhodnej postupnosti, keďže hodnoty úspešnosti štatistického testu neboli blízke hodnote 1 (viď tabuľka 4.2).
- Obrázok spolu s jeho oblasťami vykazoval značnú „nenáhodnosť“ (pozri hodnoty χ^2 testov v tabuľkách 4.3,4.6).
- Jednotlivé farebné zložky R, G, B obrázka majú hodnoty χ^2 testov do značnej miery rozdielne.
- Zmenšením skúmanej oblasti sa dosiahne približne rovnakého pomeru χ^2 testov pre jednotlivé stupne voľnosti v každej z R, G, B zložiek obrázka.

4.7 Experimentálna detekcia existencie správy

V tejto časti podrobíme LSB metódu vkladania správy do obrázkov vo formáte BMP tzv. *detekčnému útoku*. To znamená, že do rôznych typov obrázkov vložíme správu a následne sa budeme zo stego–obrázka snažiť zistiť, či v ňom je alebo nie je vložená správa (budeme predpokladať, že nevieme, že je v ňom vložená správa).

Základným detekčným útokom je tzv. párová analýza a následný χ^2 útok [14]. Stručne vysvetlíme jeho základnú myšlienku pre RGB model.

Základným predpokladom je, že v obrázku sú isté plochy rovnakých bodov. Aplikovaním LSB vkladania dôjde k nárastu použitých farieb v obrázku. Zistí sa celkový počet farieb v obrázku (U). Potom sa zistí počet takých farieb, ktoré sa od inej použitej farby v obrázku líšia v každej z RGB zložiek najviac o 1 (P). Vypočítame pomer

$$R = \frac{P}{\binom{U}{2}}.$$

Následne do analyzovaného obrázka pokusne vložíme pomocou LSB metódy správu. Veľkosť správy je potrebné určiť experimentálne. Následne pre takto upravený obrázok vypočítame hodnotu R'. V prípade, že v pôvodnom obrázku bola vložená nejaká správa, R' sa nebude od R veľmi líšiť. Ak však pôvodný obrázok správu neobsahoval, rozdiel medzi R' a R bude značný. Podľa [14] je otázna spoľahlivosť tohto algoritmu pre rôzne dĺžky správ. V [24] je uvedené, že χ^2 útok nefunguje dobre na súbory s fotografiami.

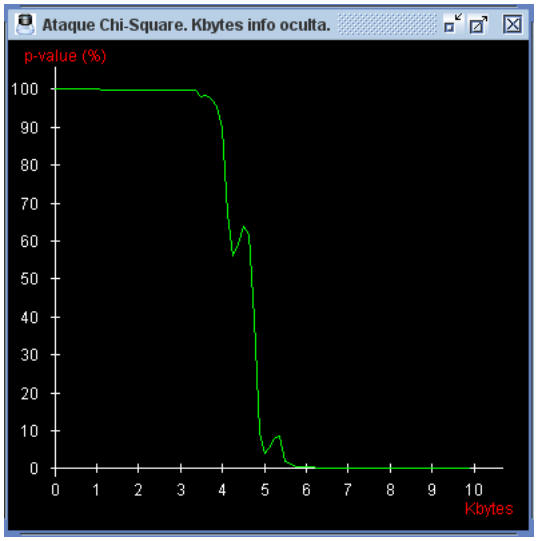
Na experimentálnu detekciu vloženej správy v súbore s obrázkom sme si zvolili voľne dostupný softvér `STEGSECRET`[1] (v0.1). Tento softvér umožňuje jednak vizuálnu analýzu obrázka — je možné zobrazovať obrázok tvorený len určitými bitovými rovinami. Taktiež je možné aplikovať na obrázok už spomínaný χ^2 test. Softvér taktiež podporuje detekciu vkladania správ do súboru s obrázkom pomocou niektorých známych steganografických nástrojov a vyhľadávanie podozrivých súborov na disku.

Pre experimenty sme si zvolili štyri obrázky rôznych druhov:

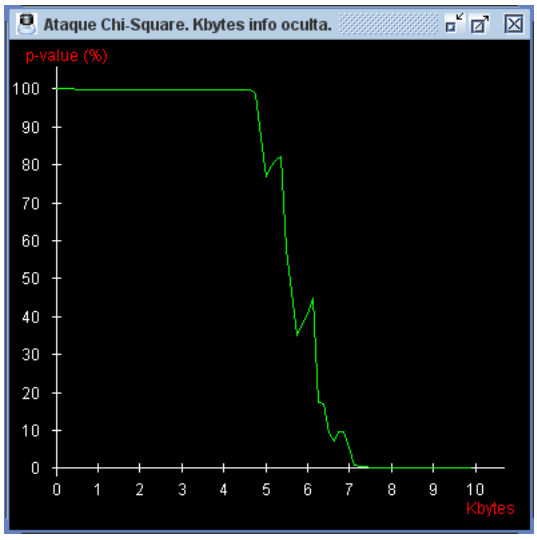
00.bmp — fotka bielej steny,

01.bmp — obrázok tvorený len červenou farbou,

[1] dostupný na `http://stegsecret.sourceforge.net`, overené 12. augusta 2011



Obr. 4.3: χ^2-test pre pôvodný obrázok 00.bmp

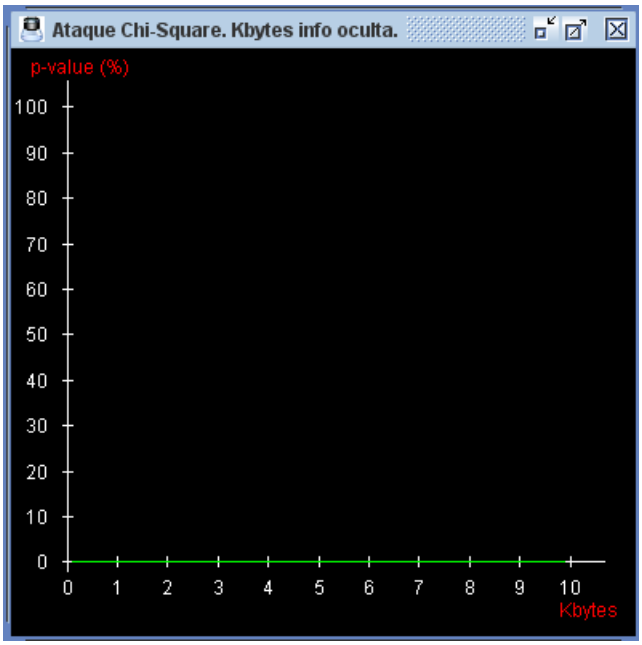


Obr. 4.4: χ^2-test pre stego-obrázok 00.bmp

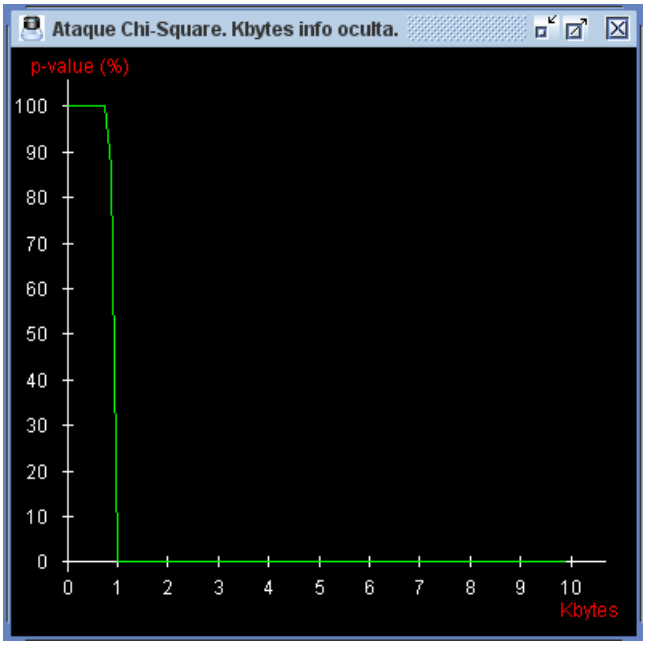
02.bmp — fotka Lomnického štítu,

04.bmp — známy portrét modelky Leny.

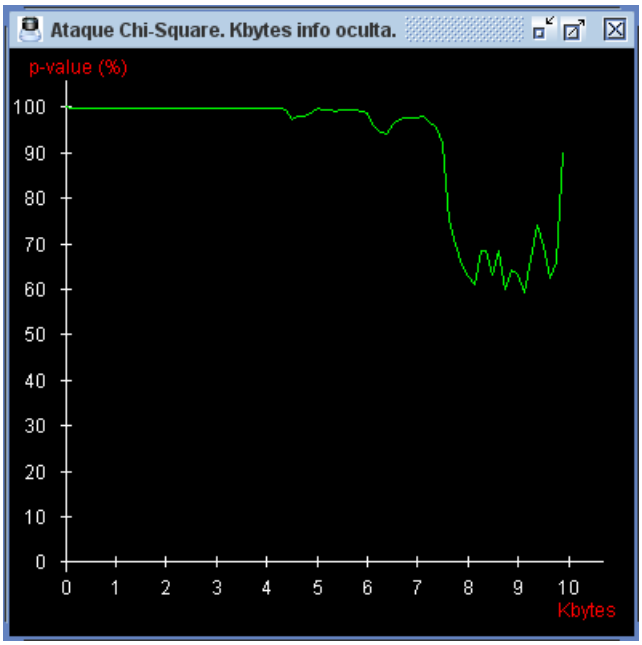
Na pôvodné obrázky, ako aj na obrázky so správami sme aplikovali χ^2 útok. Výsledky možno vidieť na obrázkoch 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 a 4.9, 4.10. Ako vidieť, potvrdilo sa, že χ^2 útok nefunguje dobre na fotografie [24].



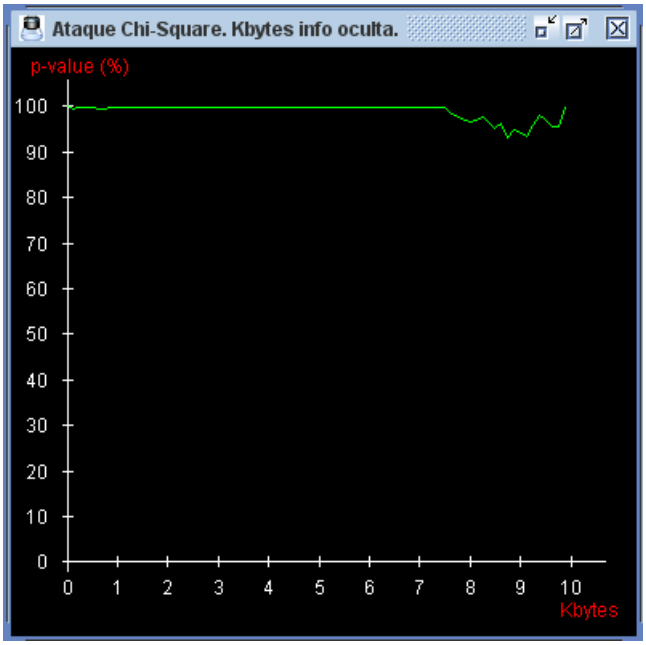
Obr. 4.5: χ^2-test pre pôvodný obrázok 01.bmp



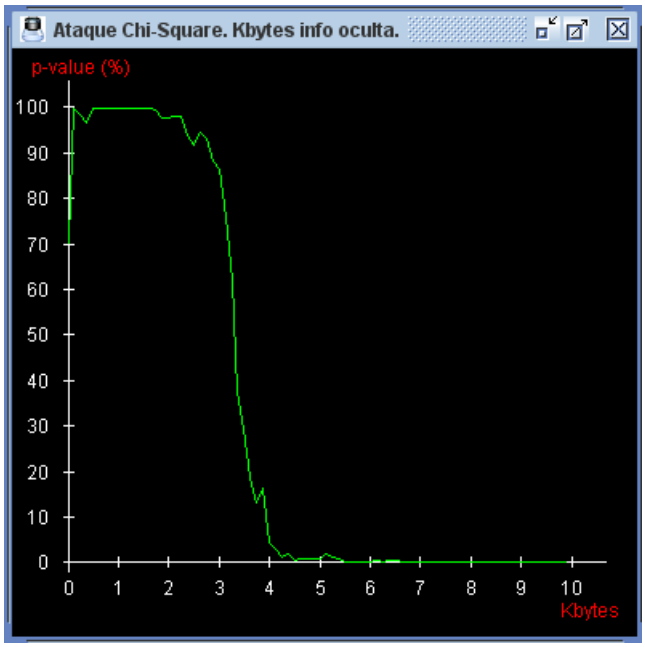
Obr. 4.6: χ^2-test pre stego-obrázok 01.bmp



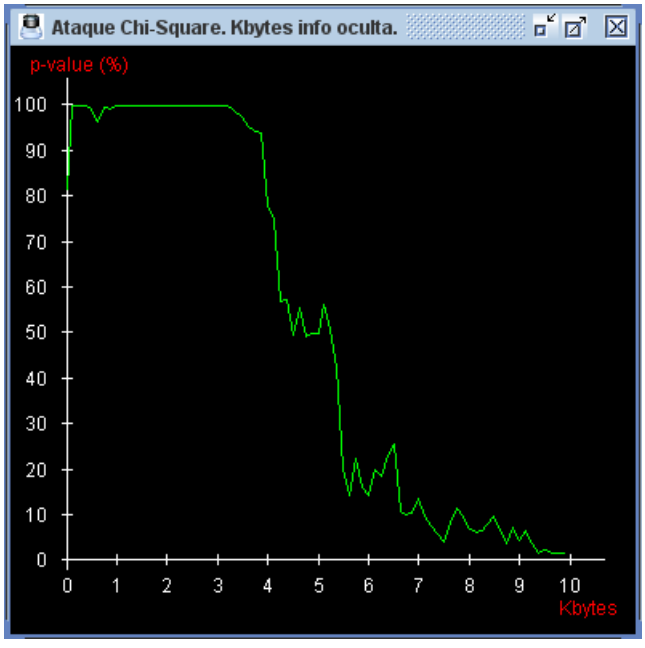
Obr. 4.7: χ^2-test pre pôvodný obrázok 02.bmp



Obr. 4.8: χ^2-test pre stego-obrázok 02.bmp



Obr. 4.9: χ^2-test pre pôvodný obrázok 04.bmp



Obr. 4.10: χ^2-test pre stego-obrázok 04.bmp

Kapitola 5

Steganografický systém založený na nekomprimovanom videu

Cieľom tohto návrhu bolo vytvoriť prototyp softvérového produktu, ktorý by bol schopný vkladať používateľom zvolený súbor (správu) do video sekvencie. Jedná sa o systém, ktorý je prototypom a ako taký má poukázať na možnosti využitia steganografie v oblasti pohyblivých obrazov. Preto nebolo potrebné vopred uvažovať rôzne súborové formáty a kódovania videa, ani rôzne steganografické techniky a algoritmy. Prototyp programu mal preukázať schopnosť vloženia a výberu správy na jednom zvolenom súborovom formáte nekomprimovaného videa pre algoritmus vkladania LSB technikou.

5.1 Motivácia k návrhu systému založeného na nekomprimovanom videu

Steganografia vo video súboroch ešte nenachádza také uplatnenie ako v prípade obrazovej steganografie. Dôvodom tejto situácie je predovšetkým nedostatok softvérových produktov z predmetnej oblasti. Niekoľko programov vzniklo už pred viacerými rokmi ako odpoveď na otázku, či je možné steganografiu vo video súboroch vôbec realizovať. Ale v súčasnosti sú tieto aplikácie zastaralé a neschopné plniť svoju funkciu z dôvodu prudkého rozvoja štandardov kódovania pohyblivého obrazu. Mnohé z pôvodných štandardov sú prekonané a už sa viac nepoužívajú. Aplikácie dostupné na trhu: `MSU Stego Video`[1] a príklad implementácie steganografického frameworku `Steganodotnet4` v jazyku C#[2].

Príklad implementácie steganografického prostredia v jazyku C# je síce voľne dostupný a modifikovateľný, ale v praxi nepoužiteľný. Jeho implementácia využíva internú knižnicu systému Windows pre prácu s videom, ktorá je zastaralá a dávno nepoužívaná. Preto tento program nie je schopný pracovať s ľubovoľnými typmi videa.

[1] dostupné na `http://www.compression.ru/video/stego_video/index_en.html` alebo `http://www.fileheap.com/software/steganography.html`, overené 12. augusta 2011

[2] dostupné na `http://www.codeproject.com/KB/security/steganodotnet4.aspx`, overené 12. augusta 2011

Na druhej strane, riešenie zásuvného modulu aplikácie `VirtualDub` (`MSU Stego Video`) má nasledovné nevýhody:

- nedostupnosť zdrojových kódov,
- nemožnosť úpravy a kontroly riešenia,
- nemožnosť rozširovania funkčnosti pridávaním podpory pre nové steganografické metódy vkladania správ do video súboru.

Naším produktom sme chceli overiť možnosť realizácie steganografie v oblasti videa. Chceli sme urobiť jednoduchý návrh a implementáciu, ktorým by sa dalo poukázať na široké možnosti v skúmanej oblasti a tak sa pripraviť na návrh steganografického systému v oblasti komprimovaného videa, o ktorom diskutujeme v kapitole 7.

5.2 Súborový formát nekomprimovaného videa

Formát video súboru je formát, v ktorom sú rôznorodé údaje (údaje opisujúce obrazovú predlohu, zvukové informácie, titulky, popisy) uložené vo forme súboru ([57]). Tieto súbory boli zavedené tak ako všetky ostatné z dôvodu potreby efektívne narábať s údajmi, ktoré reprezentujú. Samotný video súbor je taký súbor, ktorý obsahuje všetky údaje potrebné pre zobrazenie a manipulovanie s videom.

Jestvuje veľa kategórií video formátov: avi, 3gp, mov, asf, matroska,.... Pre naše účely je zaujímavý súborový formát AVI. Nielen pre jednoduchosť jeho použitia, ale aj pre jeho rozšírenosť. S veľkou rozšírenosťou úzko súvisí nenápadnosť použitia na mnohých miestach prenosu informácií.

Audio Video Interleave (AVI) je kontajnerový formát pre ukladanie obrazu a zvuku. Voľne by sa to dalo preložiť ako prekladaný zvuk s obrazom. Použila ho firma Microsoft už v operačnom systéme Windows 3.11. Video dáta boli pôvodne bez kompresie s rozlíšením 160 × 120 bodov pri 15 snímkach za sekundu, pretože procesory neboli také výkonné, aby zvládli dekompresiu v reálnom čase. Následne bol tento formát doplnený o vyššie rozlíšenie, vrátane voľby kodéra pre zníženie dátového toku. Existujú tri verzie formátu AVI, ktoré boli postupne vyvíjané a zavádzané do praxe. Na začiatku súboru je hlavička, ktorá obsahuje informácie o videu (počet snímok za sekundu, rozlíšenie, kompresia,...) a o zvuku (vzorkovacia frekvencia, počet kanálov, kompresia,...), na konci je tabuľka, v ktorej sú informácie o poradovom čísle jednotlivých snímok videa, resp. audio rámca a o ich pozíciách v súbore. Typy AVI súborov:

AVI 1.0 — umožňuje nahrávať iba do veľkosti 1GB, maximálny počet snímok je 22500, čo je asi štvrť hodiny záznamu pre 25 fps, používal sa vo Windows 3.1, už sa nepoužíva.

AVI 1.1 — rozšírenie nahrávania a indexovania do veľkosti súboru 2GB, niektoré programy riešia prekonanie tejto bariéry pomocou nahrávania do viacerých súborov.

AVI 2.0 — možná veľkosť súboru nad 2GB (obmedzená len možnosťami samotného operačného systému).

AVI sa nehodí na prenos cez Internet, ale na zachytávanie videa určeného na ďalšie spracovanie (editáciu). Je podporovaný veľkým množstvom programov a používa sa aj v iných operačných systémoch než v tých od Microsoftu. Najrozšírenejším kodérom v tejto oblasti je DV AVI, čo je AVI súbor s kompresiou DV (*Digital Video*), ktorá sa používa v digitálnych kamerách, kde je zaručená synchronizácia obrazu a zvuku.

AVI je zatiaľ najrozšírenejším kontajnerom pre ukladanie videa.[3] Podporuje väčšinu kompresií zvuku i obrazu, aj keď niektoré formáty mu robia problém. Výhodou formátu je veľká kompatibilita so všetkými operačnými systémami. Nevýhodou je nekompatibilita s novými kvalitnými formátmi zvuku a obrazu. Najčastejšie sa vyskytujúce kodeky videa pre AVI: DivX, Xvid, Cinepak, Indeo, MJPEG a DV. Najčastejšie sa vyskytujúce kodeky zvuku pre AVI: MP3, MS ADPCM a nekomprimovaný PCM.

5.3 Stegoanalýza histogramu statického obrázka

Všetky štatistické metódy stegoanalýzy obrázkov vychádzajú z nie celkom evidentného faktu, a síce, že LSB bity obrázkov nie sú náhodné. V minulosti viacerí autori nezávisle na sebe prepokladali, že LSB bity sú náhodné a preto môžu byť úspešne použité na vloženie správy. Avšak Westfeld a Pfitzmann v [81] ukázali, že tento predpoklad nie je pravdivý. Tým vlastne teoreticky umožnili odhalenie ukrytej správy.

Fakt, že LSB bity nie sú náhodné sa dá jednoduchým spôsobom použiť pri analýze obrázkov. Hlavná idea detekcie vloženej správy vychádza z toho, že pri vkladaní šifrovanej (skomprimovanej) správy sa LSB bity postupne „znáhodňujú“. Teda pri stegoanalýze obrázkov sa vlastne sleduje náhodnosť LSB bitov. Na tejto myšlienke je založená väčšina stegoanalytických metód. Asi najznámejšou stegoanalytickou metódou obrázkov, aplikovanou pri vkladaní do LSB bitov, je útok pomocou histogramu.

Prepokladajme nasledovnú situáciu. Nech N označuje obrázok (nosič) s 256 odtieňmi šedej, do ktorého chceme vložiť správu m. Nech S označuje obrázok, ktorý vznikne z N po vložení správy m do jeho LSB bitov. Analyzujme teraz N vo vzťahu k S.

Pri analýze sa snažíme zodpovedať dve základné otázky:

- Aké sú invarianty vloženia správy m. Inak povedané, aká charakteristika obrázka N zostáva zachovaná pri vložení správy m.
- Ktoré charakteristiky pôvodného obrázka N sú vložením správy zmenené. Teda čím sa líšia charakteristiky obrázkov N a S.

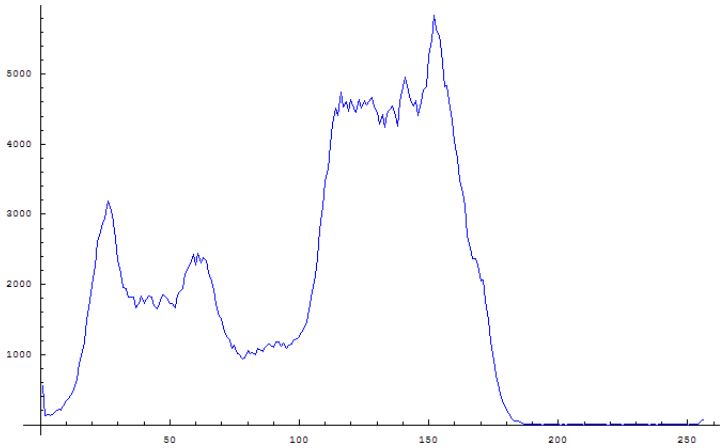
Odpovedzme najprv na prvú otázku. Prepokladáme, že nemáme o štruktúre správy m žiadnu informáciu, rovnako ani o pozíciách pixelov, do ktorých sa táto správa vkladá.

Ujasnime si však najprv, ako sa menia hodnoty farieb jednotlivých pixelov. Hodnota farby $p \in \{0,\ldots,255\}$ pixelu P sa mení v závislosti od parity p. Teda pre párne p sa hodnota p nemení alebo zvýši o jedna. Naopak pre nepárne p sa hodnota farby nemení alebo zmenší o jedna. Z tohto pohľadu teda odtiene šedej $(2i, 2i+1)$ tvoria pár farieb, ktoré sa môžu meniť len medzi sebou.

[3] hoci ho už vytláča kontajner `mp4`

Preto invariantom pri vložení ľubovoľnej správy m je početnosť pixelov s hodnotami i a $i+1$ odtieňov šedej. Pri meraní početností pixelov jednolivých farieb sa používa histogram H jednotlivých farieb obrázka. Jedná sa vlastne o vektor $H \in Z^{256}$, ktorý je zložený z početností (frekvencií) pixelov obrázka danej farebnej zložky. Teda histogram H_n nosiča N obsahuje ako svoju i–tu zložku ($H_n[i]$) počet pixelov nosiča N s farebným odtieňom i. Ako príklad možno vziať obrázok 5.1, ktorého histogram modrej zložky možno vidieť na obrázku 5.2. Pre prehľadnosť sú v uvedenom histograme jednotlivé hodnoty spojené čiarou.

Obr. 5.1: Analyzovaný obrázok.



Obr. 5.2: Histogram modrej zložky obrázka č. 5.1.

Nech H_n, H_s sú histogramy nosiča N a stego-obrázka S. Nech l označuje počet pixelov nosiča N, do ktorých bola vložená správa m s dĺžkou q bitov pre $0 \leq q \leq 1$. Zaujíma nás, ako možno na základe hodnôt $H_n[2i], H_n[2i+1]$ histogramu nosiča H_n odhadnúť hodnoty $H_s[2i], H_s[2i+1]$ histogramu nosiča H_s.

Zisťujeme teda pravdepodonosť $p_{zmenafarby}(P)$, s akou zmení konkrétny pixel P hodnotu svojej farby. Túto pravdepodobnosť možno vypočítať ako súčin pravepodobnosti toho, že správa sa vkladá do LSB bitu daného pixelu $p_{\texttt{vloženia}}(P)$ a pravdepodobnosti toho, že ho mení $p_{\texttt{zmena}}(P)$. Teda

$$p_{\texttt{zmena farby}}(P) = p_{\texttt{vloženia}}(P).p_{\texttt{zmeny}}(P).$$

Pri predpoklade, že správa je šifrovaná, a teda ju možno chápať ako sled náhodných bitov, možno pre $p_{\texttt{zmeny}}(P)$ písať

$$p_{zmeny}(P) = \frac{1}{2}.$$

Podobne, ak predpokladáme ideálny prípad ($p_{\texttt{vloženia}}(P)$ je pre každý pixel rovnaká), že $p_{\texttt{vloženia}}(P) = q$, tak pravdepodobnosť toho, že konkrétny pixel zmení hodnotu farby z $2i$ na $2i+1$ je rovná $\frac{q}{2}$. Preto je očakávaný počet pixelov, ktoré menia farbu z $2i$ na $2i+1$ rovný hodnote $\frac{q}{2}H_n[2i]$. Obdobne je počet pixelov meniacich farbu v opačnom smere rovný $\frac{q}{2}H_n[2i+1]$.

Pre párne a nepárne hodnoty histogramu nosiča H_s možno v tomto idealizovanom prípade písať:

$$H_s[2i] = \left(1 - \frac{q}{2}\right) H_n[2i] + \frac{q}{2} H_n[2i+1],$$

$$H_s[2i+1] = \frac{q}{2} H_n[2i] + \left(1 - \frac{q}{2}\right) H_n[2i+1].$$

Pre správu, ktorá sa vkladá do všetkých pixelov ($q = 1$) sú teda očakávané hodnoty $H_s[2i]$ a $H_s[2i+1]$ rovnaké. Toto možno testovať χ kvadrát testom, pomocou ktorého možno, ako sme spomínali v predošlej kapitole (na strane 52), vo všeobecnosti vyhodnocovať mieru odchýlky pozorovaných veličín od teoreticky očakávaných.

V našom prípade teda zisťujeme mieru odchýlky vzájomne zviazaných hodnôt dvoch susedných farebných odtieňov $H_s[2i]$ a $H_s[2i+1]$ od teoreticky očakávanej priemernej hodnoty $R[i] = (H_s[2i] + H_s[2i+1])/2$.

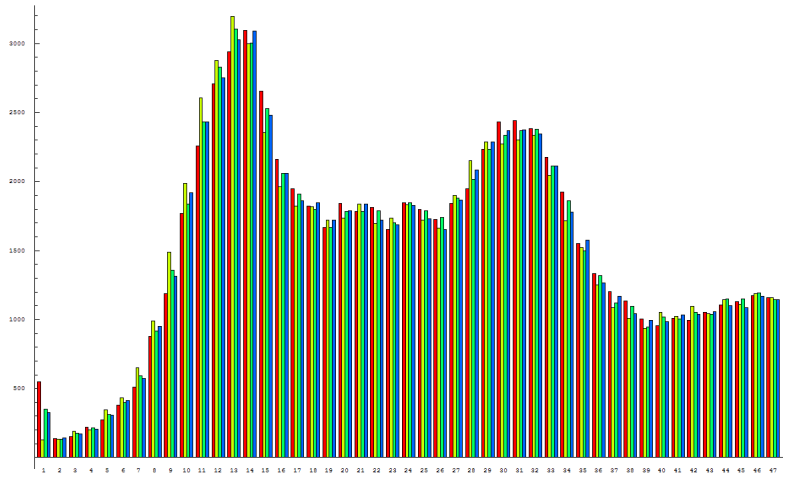
Na obrázku č. 5.3 možno vidieť spojený histogram pôvodného obrázka č. 5.1 a z neho vytvoreného stego–obrázka, ktorý obsahuje vloženú správu v LSB bite každého obrazového bodu ($q = 1$). V histograme možno vidieť, ako sa menia a vyrovnávajú frekvencie farieb $2i$ a $2i+1$ pôvodného obrázka (na obrázku sú to prvé dva stĺpce z každej štvorice) prechodom na obrázok so stegosprávou (druhé dva stĺpce štvorice — krikľavo zelená a modrá).

Bez ujmy na všeobecnosti možno pri teste použiť len párne hodnoty pixelov, keďže odchýlky $H_s[2i], H_s[2i+1]$ od očakávanej priemernej hodnoty $R[i] = (H_s[2i] + H_s[2i+1])/2$ sú zhodné.

V našom prípade má teda štatistika χ^2 testu pre párne hodnoty tvar:

$$\chi^2 = \sum_{i=0}^{k} \frac{(H_s[2i] - R[i])^2}{R[i]}, \tag{5.1}$$

kde počet stupňov voľnosti je $k = 128 - 1 = 127$ (v histograme sa nachádza 128 vzájomne zviazaných farebných párov).



Obr. 5.3: Spojený histogram pôvodného obrázka č. 5.1 a obrázka s $q = 1$.

Intuitívne veľké hodnoty χ^2 štatistiky indukujú, že rozdelenie LSB bitov nie je náhodné a teda obrázok neobsahuje stegosprávu. Naopak, malé hodnoty χ^2 štatistiky hovoria o jej existencii. V tabuľke 5.1 možno vidieť hodnoty χ^2 štatistík jednotlivých farebných zložiek (RGB) obrázka č. 5.1 v závislosti od veľkosti vloženej správy. Do každej farebnej zložky boli postupne vkladané komprimované (šifrované) správy, ktoré pokryli $0\%, 10\%, \ldots, 100\%$ všetkých obrazových bodov danej zložky.

Farba	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
B	456	411	369	338	257	176	149	106	84	56	52
G	455	414	356	333	296	227	142	102	89	59	65
R	429	340	275	265	218	159	116	99	86	71	73

Tabuľka 5.1: Hodnoty χ^2 v závislosti od veľkosti vloženej správy.

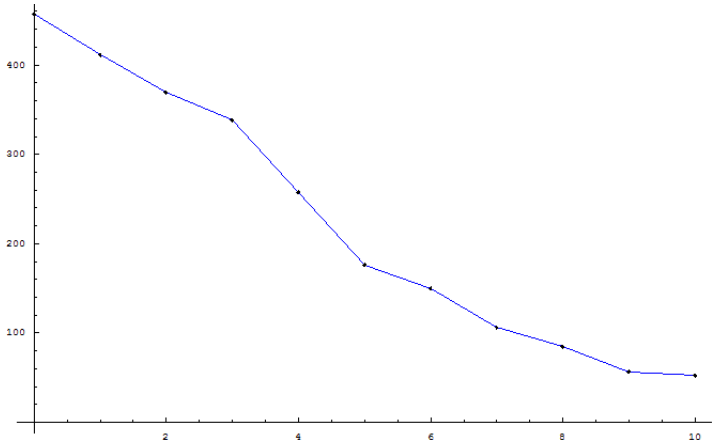
Na obrázku č. 5.4 možno vidieť vizualizáciu hodnôt modrej farebnej zložky z tabuľky č. 5.1.

Pri interpretácii výsledkov χ^2 testu sa používa distribučná funkcia χ^2 rozdelenia s príslušným stupňom voľnosti. Hodnota pravdepodobnosti, že pozorovaná veličina má rovnaké rozdelenie ako očakávaná, je daná hodnotou $1 - F_k(x)$, kde $F_k(x)$ predstavuje distribučnú funkciu χ^2 rozdelenia s k stupňami voľnosti v bode χ^2, t.j. $F_k(\chi^2)$.

V literatúre je funkcia $1 - F_k(x)$ známa tiež ako p–hodnota. Pre praktické účely sa najskôr vypočíta prahová hodnota χ^2_{prah} štatistiky pre danú hladinu významnosti α, teda $1 - F_k(\chi^2_{prah}) = \alpha$. Pre rozhodnutie o tom, či daný obrázok obsahuje stegosprávu potom stačí len porovnať hodnoty $\chi^2_{k,prah}$ a vypočítanej štatistiky χ^2. Ak platí $\chi^2 < \chi^2_{k,prah}$, potom s pravdepodobnosťou $1 - \alpha$ obsahuje analyzovaný obrázok stegosprávu. Obvykle sa pre hladinu významnosti volí hodnota $\alpha = 0,05$, čomu prislúcha prahová hodnota

$$\chi^2_{127,prah} = 154.$$

To znamená, že pri našom obrázku 5.1 dokážeme odhaliť prítomnosť stegosprávy pri pokrytí $\frac{154}{256} * 100\% \doteq 60\%$ všetkých pixelov.



Obr. 5.4: Hodnoty χ^2_{127} pre modrú zložku obrázka č. 5.1.

5.4 Stegoanalýza viacerých za sebou idúcich obrázkov

Doteraz nám nie je známy žiaden dostupný produkt, ktorý by riešil stegoanalýzu vo videosúboroch. Rôzni autori pri analýze videa používajú rovnaké techniky ako pri obrázkoch. Teda video chápu ako sled za sebou idúcich nezávislých obrázkov (angl. *frame*), a tie analyzujú technikami používanými pri obrázkoch. My sa v tejto časti zameriame na štatistickú analýzu vťahov medzi dvoma za sebou idúcimi obrázkami s ohľadom na ich LSB bity. Našou snahou je určiť zhodné charakteristiky susedných obrázkov (bez stegosprávy) a na ich základe potom navrhnúť metódy, ktoré dokážu zistiť o obrázku ďalšie informácie (napr. veľkosť vloženej správy a/alebo jej umiestenenie).

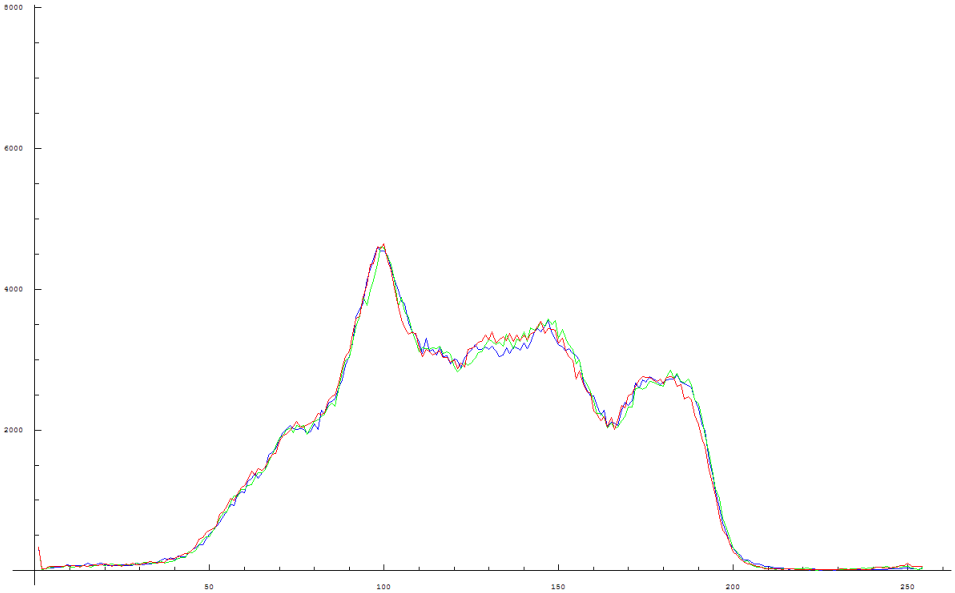
Pri určovaní zhodných charakteristík susedných obrázkov vychádzame z predpokladu, že histogramy susedných obrázkov sú približne rovnaké. Samozrejme „rovnakosť“ závisí na rýchlosti, s akou sa farebne mení snímaná scéna. Analyzovali sme preto scény rôznych typov (budova, stena, cesta a zväčšná cesta). Výsledky niektorých experimentov prezentujú obrázky 5.5 a 5.6. V nich možno vidieť histogramy modrej farby troch po sebe idúcich obrázkov budovy a cesty. Pre ďalšie prostredia (stena, cesta) sa správajú histogramy obdobne, rovnako tak pre ďalšie farby (červenú a zelenú).

Z obrázkov 5.5 a 5.6 je vidieť, že histogramy po sebe idúcich obrázkov sú približne rovnaké.

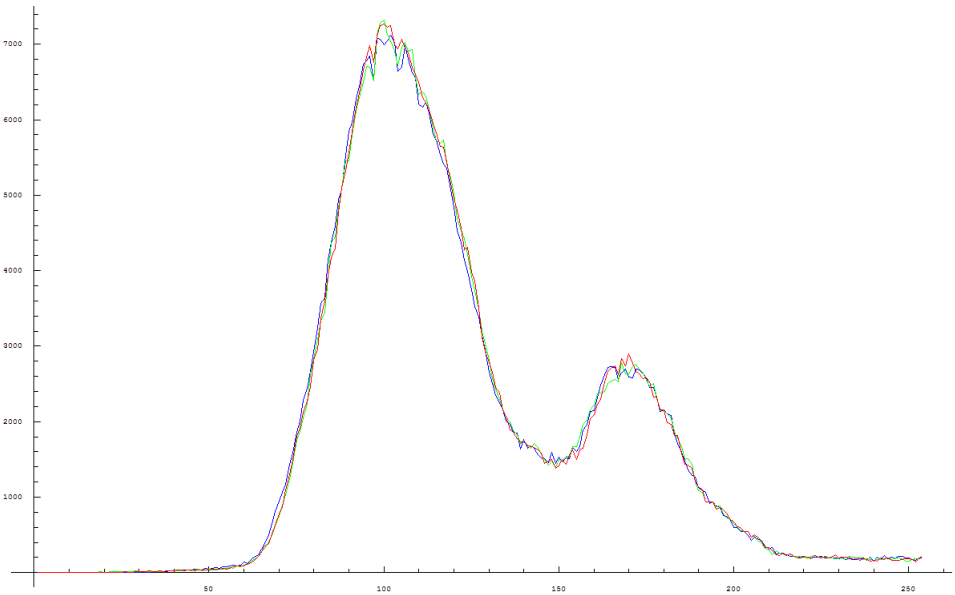
Otázne je, ako sa to prejaví pri štatistickom meraní. Podobne ako pri obrázkoch, zamerali sme sa na štatistické meranie odchýlok invariantov jednotlivých obrázkov, teda odchýlok hodnôt $R_n[i] = (H_n[2i] + H_n[2i+1])/2$ dvoch po sebe idúcich obrázkov N_1 a N_2. χ^2 štatistika má v tomto prípade (pre $k = 127$) vyjadrenie:

$$\chi^2 = \sum_{i=0}^{k} \frac{(R_2[i] - R_1[i])^2}{R_1[i]}. \tag{5.2}$$

V tabuľkách 5.2 a 5.3 možno vidieť hodnoty χ^2 štatistiky pre budovu a cestu pre dva po sebe idúce obrázky z celkového počtu 12. Teda analyzované páry obrázkov boli postupne $(1,2),(2,3),\ldots,(11,12)$.



Obr. 5.5: Histogramy troch susedných obrázkov budovy.



Obr. 5.6: Histogramy troch susedných obrázkov cesty.

Ako možno z tabuliek vidieť, všetky χ^2 štatistiky sú väčšie než prahová hodnota $\chi^2_{127,prah} = 154$. Možno teda vyvodiť nasledujúci záver: z hľadiska štatistickej zhody susedných obrázkov (bez vloženej správy) sú tieto „histogramovo“ odlišné. Keďže už samotné obrázky bez vloženej správy sú odlišné, tak aj stegoanalýza videa neprinesie žiadne ďalšie relevantné informácie oproti stegoanalýze jednotlivých obrázkov.

farba	χ^2 štatistiky										
B	696	676	949	529	592	833	500	235	407	577	665
G	662	288	668	429	410	743	1026	197	450	567	1123
R	477	734	942	410	262	525	641	244	440	407	838

Tabuľka 5.2: Hodnoty χ^2 medzi dvojicami susedných obrázkov budovy.

farba	χ^2 štatistiky										
B	935	246	374	630	282	289	960	333	343	1391	232
G	209	143	263	213	195	192	468	281	347	371	167
R	232	205	273	328	231	201	482	272	259	463	161

Tabuľka 5.3: Hodnoty χ^2 medzi dvojicami susedných obrázkov cesty.

Kapitola 6

Steganografický systém založený na súborovom formáte JPEG

Po skúsenostiach s implementáciou steganografických systémov založených na nekomprimovaných obrazových údajoch sme pristúpili k realizácii systému pracujúceho s obrazom komprimovaným technológiou JPEG. Zamerali sme sa na tento typ súboru, pretože táto technika kompresie je využívaná pri kódovaní pohyblivého obrazu v aktuálne používaných video štandardoch rodiny MPEG. Časti 6.1 a 6.2 boli vypracované v úzkej spolupráci s autorom [56].

6.1 Úvod do JPEG

V osemdesiatych rokoch posledného storočia minulého tisícročia vznikol silnejúci tlak viacerých veľkých korporácií o ustanovenie nového obrazového štandardu, ktorý by spĺňal nasledovné požiadavky:

- nezávislosť na grafickom zobrazovacom zariadení,
- minimalizovanie veľkosti pomocou kódovacích techník kompresie údajov,
- efektívnosť implementácie v hardvéri, firmvéri a softvéri,
- vhodnosť pre fotografie (nielen čiernobiele, ale aj farebné).

Preto bola v roku 1982 vyhlásená súťaž o novú stratovú komprimačnú technológiu. Súťaž prebiehala pod dohľadom pracovnej skupiny JPEG (z angl. *Joint Photographics Experts Group*) v rámci medzinárodnej štandardizačnej organizácie ISO.

O päť rokov neskôr jestvovalo dvanásť návrhov pre kompresiu bitmapovo orientovaného obrazu. O ďalšie dva roky bol konečne dostupný prvý návrh nového komprimačného štandardu doteraz označovaného podľa pracovnej skupiny `JPEG`.

Nový algoritmus sa skladal z troch základných blokov:

1. z prevodu farebného modelu používaného zariadeniami (RGB) do YC_bC_r modelu,
2. z transformácie z priestorovej oblasti do frekvenčnej (tzv. DCT transformácia), a
3. z bezstratového kódovania získaných DCT koeficientov.

V nasledujúcom roku (1990) bol návrh algoritmu ukončený, a dva roky nato konečne prijatý ako štandard č. ISO/IEC 10918-1.

Jednou z najdôležitejších požiadaviek kladených na nový štandard bola efektivita potrebného pamäťového miesta na uchovanie grafickej informácie. Preto sa v návrhu použili metódy stratovej kompresie. Tie sú založené na skutočnosti, že cieľový účastník schémy nie je schopný vnímať všetky informácie predkladaného grafického objektu. Ľudský vizuálny systém je v oveľa väčšej miere citlivý na jas ako na farebnú informáciu. Preto je možné v istej miere (ktorá sa zisťovala na základe experimentov) farebnú informáciu zredukovať bez toho, aby človek pozoroval degradáciu obrazu. Tým je možné podstatným spôsobom znížiť množstvo údajov potrebných pre uchovanie grafickej informácie.

Experimentami sa zistilo, že pri použití 15 až 25 násobného faktoru kompresie neprichádza k viditeľnej degradácii pôvodnej grafickej informácie. Je však na používateľovi, aby si zvolil kompresný pomer podľa svojich potrieb alebo požiadaviek. Komprimačný pomer je možné zvyšovať až dovtedy, kým to používateľovi vyhovuje [76].

Ukazuje sa, že použitie 50 až 100 násobného faktora kompresie pri istých typoch obrázkoch nevedie k podstatnej degradácii originálneho obrazu. Týka sa to napríklad takých typov obrazu, v ktorých sa nachádzajú plynulé prechody (nie ostré hrany alebo monotónnej plochy) medzi jednotlivými farebnými komponentami. Pri takých typoch obrázkov, ktoré obsahujú ostré hrany alebo monotónne farebné plochy s malou farebnou hĺbkou je technológia JPEG najmenej účinná. Preto je dôležité vedieť, na aký typ obrazu sa môže použiť. To vychádza z definície požiadaviek štandardu; jedná sa predovšetkým o zachytený obraz reality (fotografie), ktorý má veľkú farebnú hĺbku a spojité prechody medzi jednotlivými objektami popredia a pozadia. Taktiež je možné efektívne používať štandard JPEG na kódovanie obrazu v odtieňoch šedej. Určite však nie je určený pre uchovávanie obrázkov, na ktorých sa nachádzajú ostré hrany (písmo).

Ešte sa pre ujasnenie terminológie vrátime k názvu komprimačnej metódy a k označeniu súborového formátu, v ktorom je natívne implementovaná. Všeobecne používané označenie `JPEG` je, ako sme už boli spomínali, označením pracovnej skupiny, ktorá mala na starosti vývoj nového štandardu. Od názvu tejto skupiny sa odvíja aj samotný názov komprimačnej techniky. Označuje sa ním algoritmus, ktorý je výstupom práce uvedenej skupiny. Technicky správny názov súborového formátu, ktorému sa v tejto časti venujeme, nie je `JPEG`, ale `JFIF`. Samotná komprimačná technika `JPEG` totiž nijako nešpecifikuje fyzické charakteristiky uchovávania grafickej informácie v nejakom súborovom formáte. Uvedený komprimačný algoritmus sa využíva totiž nielen v súborovom formáte `jpg`, ale aj v iných grafických (alebo multimediálnych) formátoch, napríklad `pdf`, `ps`, `mov`, `avi`, `mkv`, `mp4`, `tiff`,....

Aj keď bola základná verzia algoritmu dokončená v roku 1992, odvtedy vzniklo viacero rozšírení, ktoré majú rôzne oblasti zamerania. Popis niektorých súvisiacich štandardov sa nachádza v tabuľke č. 6.1.

Okrem stratovej kompresie je možné využiť aj techniku bezstratovej kompresie poskytovanú špeciálne upraveným algoritmom JPEG. Pre obrázky v dvoch farbách (predovšetkým v čiernej a bielej) bola vyvinutá tiež špeciálne derivácia algoritmu označovaná názvom `JBIG`. Napokon v roku 2000 boli predošlé štandardy upravené tak, že miesto diskrétnej kosínusovej transformácie sa môže používať aj tzv. vlnková (z angl. *wavelet*) transformácia.

Tabuľka 6.1: Rôzne štandardy `JPEG`.

popis	zodpovedajúci štandard
(bez)stratový JPEG	ITU-T T.81, ISO/IEC IS 10918-1
rozšírený JPEG	ITU-T T.84
vylepšený bezstratový JPEG-LS	ITU-T T.87, ISO/IEC IS 14495-1
čierno–biely JBIG	ITU-T T.82, ISO/IEC IS 11544-1
JPEG–2000 (náhrada za JPEG-LS)	ITU-T T.800, ISO/IEC IS 15444-1
rozšírený JPEG–2000	ITU-T T.801

Súbor `jpg`, natívne pracujúci s údajmi spracovávanými štandardami rodiny `JPEG`, používa na uchovávanie údajov pamäťový model tzv. `big endian`. V tomto pamäťovom modeli sa bajty s vyššou váhou nachádzajú na nižších pamäťových adresách. V duálnom pamäťovom modeli, označovanom `little endian`, sa bajty s vyššou váhou nachádzajú na vyšších pamäťových adresách. To znamená, že v modeli používanom súborom `jpg` sa hodnoty interpretujú v prirodzenom poradí, t.j. tak, ako sú zapísané. Ak sa v súbore nachádza hodnota 12345678_{16}, je to číslo 12345678 zapísané v šestnástkovej sústave, teda v desiatkovej vyjadruje hodnotu 305419896. V pamäťovom modeli `little endian` sa reťazec šestnástkových cifier 12345678_{16} interpretuje ako číslo 78563412_{16}, teda v desiatkovej sústave sa jedná o hodnotu 2018915346.

Poznámka 7. *Keďže sa pamäťové modely vzťahujú na adresáciu najmenšej pamäťovej jednotky (bajtu), je potrebné si uvedomiť, že v šestnáskovej sústave je jeden bajt vyjadrený ôsmymi bitmi, čo zodpovedá dvom cifrám v šestnástkovej sústave.*

Súbory typu `jpg` obsahujú údaje štruktúrované do tzv. *segmentov*. Každý segment začína *značkou*. Značka je tvorená začiatočným bajtom hodnoty $0xFF$, za ktorým nasleduje identifikátor typu značky. Prefix $0xFF$ odlišuje údaje od značiek. Ak sa v údajoch vyskytuje hodnota $0xFF$, je kódovaná pomocou dvojice $0xFF00$.

Niektoré značky sú iba dvojbajtové (teda prefix a identifikátor). Za väčšinou značiek sa však nachádza dĺžka segmentu uložená v dvoch bajtoch. Tým je dĺžka jedného segmentu obmedzená na $65535 - 2 = 65533$ bajtov, pretože dva bajty indikujúce dĺžku sa započítavajú do celkovej dĺžky segmentu.

Hodnota $0xFF$ plní okrem sémantickej funkcie aj funkciu synchronizačnú. Ak sa pri prenose súboru transportným kanálom údaje poškodia, dekodér sa môže synchronizovať pri najbližšej hodnote $0xFF$. Preto sa táto hodnota vkladá do údajovej časti obrázka. Nakoľko plní aj funkciu významovú, nesmie sa pre účely synchronizácie nachádzať osamotene (nedala by sa odlíšiť jej funkcia synchronizácie od definovania značiek). Z tohto dôvodu musí byť aj v údajovej časti nasledovaná niektorým zo synchronizačných značkových identifikátorov.

Striktne vzaté, `jpg` súbory nemajú záhlavie. Hneď na začiatku obsahujú značku `SOI` (angl. *Start Of Image*), ktorá má hodnotu $0xFFD8$. Túto značku musí obsahovať každý `jpg` súbor.

Značka `SOI` je považovaná za hlavičku súboru z dôvodov konzistentnej terminológie vzhľadom na iné typy súborov.

Hlavička súboru je nasledovaná ďalšími segmentami. Mal by sa tam nachádzať tzv. *aplikačný segment*, v ktorom sa nachádzajú špecifické informácie pre apliáciu, ktorá spracúva daný súbor. `JFIF` súbory používajú značku `APP0` ($0xFFE0$), a súbory s `EXIF` formátom zasa značku `APP1` ($0xFFE1$). Špeciálne typy súborov (napríklad použitie aritmetického kódovania miesto Huffmanovho) používajú ďalšie aplikačné značky.

Segment `APP0` obsahuje dva bajty informujúce o dĺžke, identifikačný reťazec „`JFIF`“ ukončený nulovým znakom, identifikátor verzie a subverzie (každé v osobitnom bajte) a základné informácie o obrázku:

- jednotky zobrazenia obrázka na zobrazovacom zariadení:

  $0x00$ — nie sú zadané žiadne jednotky,

  $0x01$ — rozmery sa uvádzajú v palcoch (`DPI`, angl. *Dots Per Inch*) alebo

  $0x02$ — rozmery sú udané v centimetroch (`DPCM`, angl. *Dots Per CM*);

- šírka obrázka (2 bajty),
- výška obrázka (2 bajty),
- šírka náhľadu na obrázok (1 bajt),
- výška náhľadu na obrázok (1 bajt),
- paleta farieb použitých na vytvorenie náhľadového obrázka.

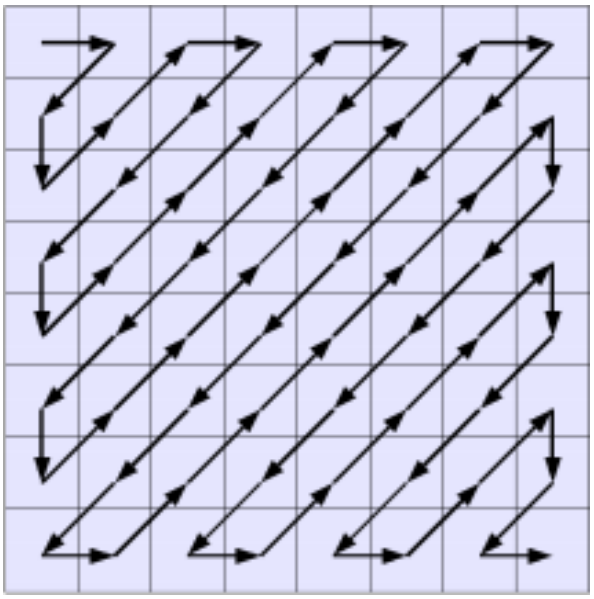
Popis jednotlivých doteraz spomenutých segmentov (`SOI` a `APP0`) sumarizuje tabuľka č. 6.2. Ostatné nevyhnutne prítomné segmenty v `jpg` súboroch sú uvedené v tabuľke č. 6.3.

Tabuľka 6.2: Segmenty `SOI` a `APP0` [56].

segment	názov	hodnota	popis	veľkosť v bajtoch
SOI	Start Of Image	0xFFD8	identifikuje začiatok súboru	2
APP0	Application Marker	0xFFE0	identifikuje súbor JFIF	2
			dĺžka segmentu APP0	2
			nulou ukončený reťazec „JFIF“	5
			číslo verzie a subverzie	2
			špecifikácia rozlíšenia	1
			horizontálne rozlíšenie	2
			vertikálne rozlíšenie	2
			šírka náhľadového obr.	1
			výška náhľadového obr.	1
			farebná paleta RGB	3*počet

Tabuľka 6.3: Segmenty nevyhnutne prítomné v súboroch `jpg`.

segment	názov	hodnota	popis
DQT	Define Quantization Table	`0xFFDB`	kvantizačné tabuľky
DHT	Define Huffman Table	`0xFFC4`	Huffmanove tabuľky
DRI	Define Restart Interoperability	`0xFFDD`	synchronizácia
SOF	Start Of Frame	`0xFFC0`	nastavenia obrázka
SOS	Start Of Scan	`0xFFDA`	nastavenia komponentov
EOI	End Of Image	`0xFFD9`	koniec údajov



Obr. 6.1: „Cik–cak“ usporiadanie DCT koeficientov.

Tabuľky kvantizačných koeficientov sú špecifikované značkou `DQT` s hodnotou $0xFFDB$. Za dĺžkou segmentu sa nachádza bajt obsahujúci dve informácie. V prvej polovici je uložená presnosť bitovej reprezentácie jednotlivých DCT koeficientov, a v druhej identifikátor tabuľky, nakoľko v jednom súbore môže byť až 16 tabuliek kvantizačných koeficientov. Zvyknú sa ale používať iba maximálne tri:

1. pre jasovú zložku Y a
2. pre chrominančné zložky spektra (C_b a C_r).

V praxi sa stretneme prevažne s použitím dvoch tabuliek, pretože tabuľka pre farebné zložky býva totožná. Potom nasleduje pole 64 prvkov jednotlivých DCT koeficientov. Každý má veľkosť buď 8 alebo 16 bitov (v závislosti od nastavenej presnosti v predchádzajúcom poli). Jednotlivé elementy tabuľky sú uložené v tzv. *cik–cak* postupnosti (viď obrázok č. 6.1). Všetky tabuľky sa nachádzajú v jednom segmente hneď za sebou.

Okrem kvantizačných tabuliek DCT koeficientov sa v súbore musia nachádzať tabuľky kódových slov pre dekódovanie DCT koeficientov zakódovaných Huffmanovým algoritmom.

Celkovo sú prítomné štyri tabuľky pre:

1. absolútny (DC) koeficient jasovej zložky,
2. ostatné (AC) koeficienty jasovej zložky,
3. absolútny koeficient farebných zložiek a
4. ostatné koeficienty farebných zložiek.

Tabuľky sú uložené za sebou a majú nasledovnú štruktúru:

- jeden bajt hlavičky pre identifikáciu typu tabuľky,
- pole bajtov číslované od indexu 1 po 16 (vrátane), ktoré obsahuje na každej pozícii počet kódových slov zodpovedajúcich dĺžke, ktorú reprezentuje daný index; suma jednotlivých počtov nesmie presiahnuť hodnotu 256;
- nasleduje pole, v ktorom sú za sebou uložené jednotlivé kódové slová od najkratšieho po najdlhšie.

Sumarizácia segmentov obsahujúcich jednotlivé tabuľky sa nachádza v tabuľke č. 6.4. Zvyšné segmenty sú popísané v tabuľke č. 6.5.

Tabuľka 6.4: Segmenty `DQT` a `DHT` [56].

segment	názov	hodnota	popis	veľkosť
DQT	Define Quantization Table	`0xffdb`	začiatok segmentu koeficientov kvantizačnej tabuľky	2 bajty
			dĺžka segmentu DQT	2 bajty
			číslo kvantizačnej tabuľky	4 bity
			počet bitov na koeficient	4 bity
			hodnoty jednotlivých koeficientov	64 bajtov
DHT	Define Huffman Table	`0xffc4`	začiatok segmentu Huffmanovho kódovania	2 bajty
			dĺžka segmentu DHT	2 bajty
			index tabuľky AC/DC	1 bajt
			počty kódových slov	16 bajtov
			jednotlivé kódové slová	variabilné

Základný štandard algoritmu `JPEG` umožňuje výber zo štyroch spôsobov kódovania obrazu. Popis rozdielov medzi jednotlivými módmi sa nachádza v tabuľke č. 6.6. Podrobný popis je k dispozícii napríklad v [1].

Sekvenčný mód — disponuje viacerými spôsobmi spracovania údajov. Jednotlivé možnosti sa odlišujú 8 alebo 12 bitovou reprezentáciou jednotlivých obrazových bodov vstupu. Ďalšie rozdiely sa nachádzajú v oblasti diskrétnej kosínusovej transformácie (môže byť základná alebo rozšírená) a v oblasti finálneho kódovania DCT koeficientov (štandardne Huffmanove alebo aritmetické). Jednotlivé možnosti znázorňuje obrázok č. 6.2.

Tabuľka 6.5: Segmenty `SOF`, `SOS` a `EOI` [56].

segment	názov	hodnota	popis	veľkosť
SOF	Start Of Frame	`0xffc0`	špecifikácia tvaru obrazových dát	2 bajty
			dĺžka segmentu	2 bajty
			presnosť v bitoch	1 bajt
			komponent Y	2 bajty
			komponent X	2 bajty
			počet komponentov	1 bajt
			index komponenty	1 bajt
			vzorkovacie faktory	1 bajt
			index kvantizačnej tabuľky	1 bajt
SOS	Start Of Scan	`0xffda`	začiatok vlastných obrazových dát	2 bajty
			dĺžka segmetu	2 bajty
			počet komponentov	1 bajt
			začiatok spektrálneho výberu	1 bajt
			koniec spektrálneho výberu	1 bajt
			index komponenty	1 bajt
			číslo AC/DC tabuľky	1 bajt
			najvyšší a najnižší bit	1 bajt
EOI	End Of Image	`0xffd9`	špecifikuje koniec súboru	2 bajty

Tabuľka 6.6: Základné spôsoby kódovania štandardu `JPEG`.

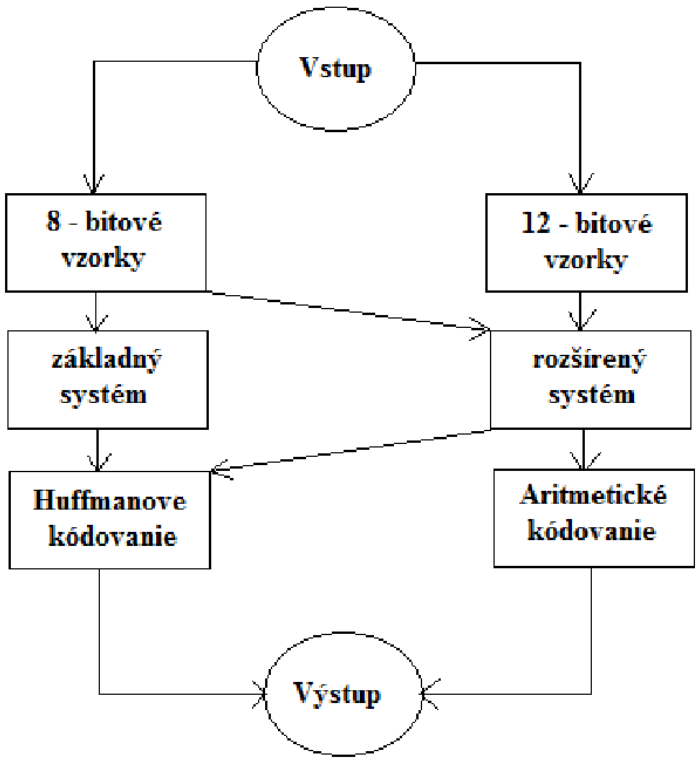
názov módu	kódovanie	charakteristika
sekvenčný	stratové	najmenej náročné na pamäť, najpoužívanejšie
progresívny	stratové	viac náročné na pamäť, určené pre prenos obrázka po sieti
bezstratový	nestratové	predikčné kódovanie, nie príliš známe ani používané/podporované, kompresný pomer 1:2
hierarchický	stratové	viac rozlíšení v jednej snímke, rýchle náhľady, podpora zobrazenia, tlače

Progresívny mód — umožňuje dekódovanie obrázka po bitových vrstvách alebo po frekvenčných kvantách. Je možné obe techniky kombinovať, teda dekódovať obrázok v jednotlivých bitových vrstvách (od najviac významnej bitovej roviny po najmenej významnú) s frekvenčnými kvantami (od nízkych frekvencií až po najvyššie). Zobrazovanie takto dekódovaného obrázka sa deje tak, že do obrazu sa v každej iterácii vkladajú stále väčšie detaily. Tento mód sa využíva predovšetkým pri prenosoch s obmedzenou komunikačnou linkou. Je výhodný napríklad pre pomalé internetové spojenia. Obrázok sa najprv vykresľuje rozmazane, a postupne sa vyostruje.

Bezstratový mód — na kódovanie grafickej informácie sa nepoužíva diskrétna kosínusová transformácia, ale jednoduchá pulzná modulácia. Hodnoty jednotlivých obrazových bodov sú kódované pomocou predikcie takým spôsobom, že nasledujúca hodnota obra-

zového bodu sa odhaduje na základe predošlého (prípadne okolitých susedov). Výsledné hodnoty sa získajú odpočítaním skutočnej a predikovanej hodnoty. Tie sa ďalej spracujú štandardným spôsobom — pomocou Huffmanovho alebo aritmetického kódovania. V tomto móde sa pôvodný dátový súbor zmenší približne na polovicu. Prehľadné spracovanie informácií o tomto móde viď napríklad v [59].

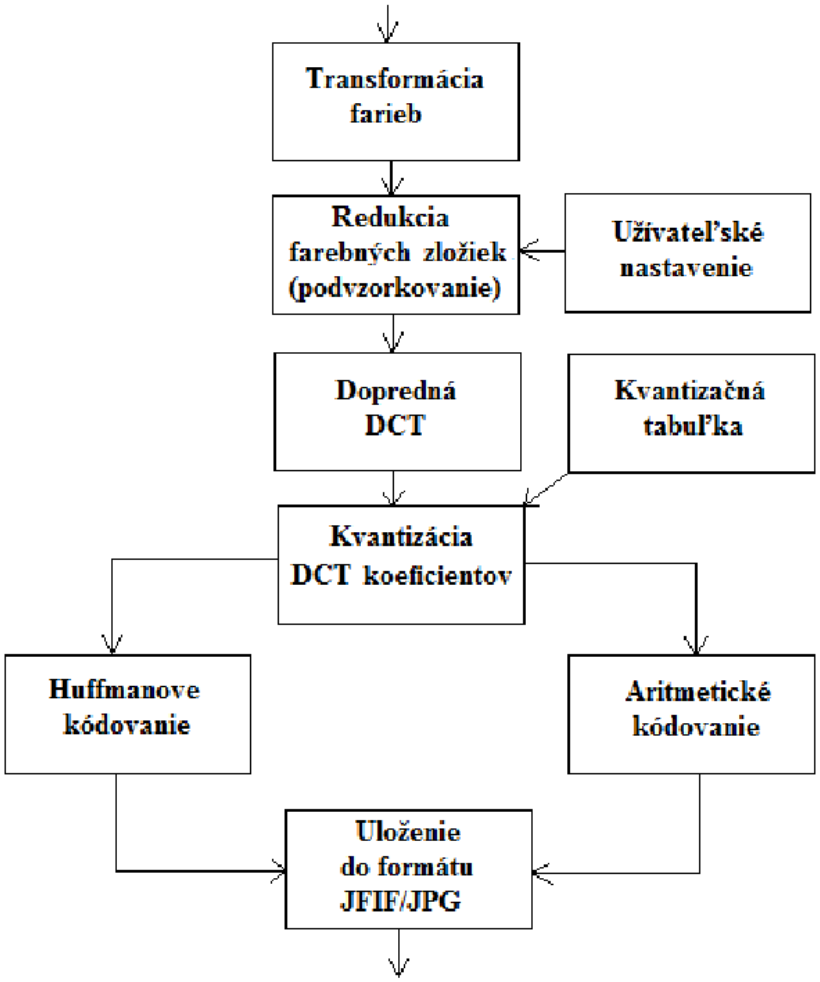
Hierarchický mód — na kódovanie výsledného obrázka sa používa sada obrázkov, ktoré vzniknú podvzorkovaním pôvodného obrázka. Každý obrázok vznikne s iným nastavením rozlíšenia. Začína sa najmenším rozlíšením, a postupuje sa k vyšším detailom. Jedná sa o iteračný proces (de)kódovania. V každom kroku tohto procesu sa vstupom predikcie nasledujúceho stáva výstup predošlého kroku. Znovu, ako v predošlom prípade, sa kódujú rozdiely predikcie oproti skutočným hodnotám. Je to z toho dôvodu, že rozdiely majú vo všeobecnosti menšie absolútne hodnoty než pôvodné dáta, takže na ich kódovanie je potrebné použiť menej bitov. Tým sa zlepšuje efektivita komprimačného pomeru.



Obr. 6.2: Procesný tok sekvenčného módu kompresie štandardu JPEG [56].

Náš steganografický systém navrhujeme pre základný (sekvenčný) mód JPEG kompresie s osem bitovými vstupnými charakteristikami obrazových bodov. Na transformáciu do frekvenčnej oblasti využívame štandardnú kosínusovú diskrétnu transformáciu a výsledné koeficienty kódujeme pomocou Huffmanovho algoritmu.

Zvolili sme si tento mód činnosti kvôli tomu, že je jediným, ktorý musí podľa JPEG štandardu implementovať každý (de)kodér. Bezstratový, progresívny a hierarchický sú iba doplnkovými módmi, ktoré nemusí (de)kodér podporovať.



Obr. 6.3: Schematické znázornenie spracovania údajov pri základnom sekvenčnom komprimačnom algoritme [56].

Schému spracovania údajov (podľa štandardu) v našom steganografickom systéme možno vidieť na obrázku č. 6.3. Ako si možno všimnúť (viď obrázok č. 6.3), stratová kompresia štandardu `JPEG` predstavuje istú postupnosť operacií. Ide o funkčné bloky vykonávajúce transformáciu farieb, podvzorkovanie, diskrétnu kosínusovú transformáciu, kvantizáciu DCT koeficientov a konečné zakódovanie kvantizovaných koeficientov. Jednotlivé kroky kompresie sú pre potreby našej práce popísané v časti 2.2. Bližší popis celého návrhu štandardu sa nachádza v [79].

6.2 Návrh steganografického systému

Ako bude ukázané v časti 7.2, jedným z hlavných cieľov bezpečnosti steganografických systémov je vkladať informáciu tak, aby nebolo možné odlíšiť médium obsahujúce steganografickú informáciu od čistého média [2].

Úspešnosť detekcie steganografickej informácie pri technikách modifikujúcich originálny dátový nosič koreluje s veľkosťou vloženej správy. Súvisí to s mierou zmeny pôvodného nosiča.

Čím viac zmien sa uskutoční, tým viac sa steganografické a čisté médium líšia. Ak tieto zmeny nie sú v súlade so štatistickými vlastnosťami pôvodného typu údajov, pravdepodobnosť nájdenia odlišujúcej štatistiky narastá. Preto je cieľom viacerých steganografických systémov minimalizovanie zmien pôvodného nosiča. To je aj naším cieľom.

Voľne (alebo aj komerčne) dostupné steganografické systémy[1] používajú na prenos vloženej informácie techniku modifikácie najmenej významných bitov [25]. Tá je vynikajúca vzhľadom na poskytnutie maximálnej kapacity,[2] ale v procese vkladania informácie modifikujú veľké množstvo pôvodných najmenej významných bitov.

Kvôli zachovaniu vlastnosti dôvernosti steganografického systému sa často vkladaná informácia predspracuje procesom šifrovania. Nakoľko výstupom vhodne zvoleného šifrovacieho algoritmu je pseudonáhodná postupnosť, ktorá má charakteristiku rovnomerného rozdelenia núl a jednotiek v zodpovedajúcom binárnom rozvoji, uvedené verejne dostupné steganografické systémy s pravdepodobnosťou 1/2 modifikujú každý použitý najmenej významný bit. Dôsledkom toho je možnosť jednoduchej detekcie použitia tejto metódy.

Preto sme sa pokúsili implementovať steganografický systém podľa návrhu [87]. Ten je založený na minimalizácii zásahov do pôvodného média. V nasledujúcom texte, ak nebude uvedené inak, považujeme za mieru efektivity minimalizáciu modifikácie originálnych údajov, do ktorých sa vkladá správa. Meranie tejto minimalizácie robíme na základe počtu informačných bitov, ktoré nám v nosiči spôsobia minimálne jednu zmenu pôvodných bitov nosiča.

Priamym dôsledkom takto navrhnutého steganografického systému je zvýšenie celkovej úrovne nedetegovateľnosti, a tým bezpečnosti implementovaného systému. Na takto navrhnutý steganografický systém nie je možné použiť tradičné postupy stegoanalýzy, ako ukážeme v časti 6.3.

Minimalizácia modifikovaných častí (bitov) pôvodného média sa dosahuje na základe vhodne zvoleného kódovania vkladanej informácie.

V steganografickom systéme založenom na modifikácii najmenej významných bitov (LSB) DCT koeficientov je (v závislosti od veľkosti vkladanej správy) veľké množstvo LSB bitov nevyužitých. Kódovaním jedného informačného bitu viacerými LSB bitmi pôvodného nosiča dokážeme znížiť potrebné množstvo vykonaných zmien.

Ako sme už spomínali v predošlom texte, predpokladáme vkladanie rovnomerne rozdelenej binárnej postupnosti údajov. V takto definovanom modeli sa jedna zmena prejaví na dvojici bitov. Pri meraní efektívnosti počtom informačných bitov spôsobujúcich jednu zmenu v nosiči, potom je v tomto prípade koeficient efektívnosti systému rovný dvom.

Ak by sme sa pokúsili vložiť dva informačné bity m_1 a m_2 do troch bitov nosiča x_1, x_2 a x_3 tak, aby sme urobili maximálne jednu zmenu v uvedených troch bitoch nosiča, môžu nastať tieto situácie:

[1] napríklad `JSteg`, `JP Hide & Seek` alebo `Outguess`

[2] ak využijeme najmenej významný bit každého bajtu, dostaneme kapacitu rovnú jednej osmine veľkosti pôvodných údajov

1. nevykonáme žiadnu zmenu, ak $m_1 = x_1 \oplus x_3 \wedge m_2 = x_2 \oplus x_3$,
2. zmeníme bitové miesto x_1, ak $m_1 \neq x_1 \oplus x_3 \wedge m_2 = x_2 \oplus x_3$,
3. zmeníme bitové miesto x_2, ak $m_1 = x_1 \oplus x_3 \wedge m_2 \neq x_2 \oplus x_3$,
4. zmeníme bitové miesto x_3, ak $m_1 \neq x_1 \oplus x_3 \wedge m_2 \neq x_2 \oplus x_3$.

Podľa takto zvoleného kódovania sa pri vkladaní dvoch informačných bitov do troch bitov nosiča nezmení nikdy viac než jeden bit pôvodného nosiča.

Po zvážení teoretického modelu predstaveného v práci [87] sme pre náš steganografický systém použili Hammingov kód. Jeho implementácia je jednoduchá a veľmi efektívna (vzhľadom na výkon). Pre názornosť prikladáme ukážkový príklad.

Na (de)kódovanie troch bitov správy použijeme sedem najmenej významných bitov pôvodného nosiča. Nech

$$H_c = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

je kontrolnou maticou (7,4) Hammingovho kódu. Stĺpce kontrolnej matice tohto kódu sú zvolené tak, aby boli binárnou reprezentáciou (odhora nadol) indexu stĺpca (stĺpce sú číslované od 1). Pre nami (pre ukážku) zvolený informačný vektor $s = 101$ a vektor tvorený LSB bitmi originálneho nosiča $x' = 1001010_2$ určíme poradové číslo bitu vektora x', ktoré je nutné zmeniť, aby sa vektor x' stal kódovým slovom zodpovedajúcim informačnému vektoru m. Označme symbolom E poradové číslo bitu vektora x', v ktorom urobíme maximálne jednu zmenu:

$$E = (H_c \,.\, x'^T) \oplus s = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Vektor $E = 110_2$ je šiestym stĺpcom kontrolnej matice H_c. Preto práve modifikáciou šiesteho bitu vektora x' dostaneme kódové slovo $x = 1001000_2$, ktorého dekódovaním získame pôvodný informačný vektor:

$$H_c \,.\, (x)^T = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = s^T$$

Je zrejmé, že jedine v prípade, ak $H_c.x'^T = s^T$, nemení sa žiaden bit vektora x'. Vo všetkých ďalších situáciách je nutné modifikovať práve jeden bit vektora tvoreného pôvodnými bitmi nosiča. Pomocou 7 bitov nosiča dokážeme vkladať tri informačné bity tak, aby sme spôsobili maximálne zmenu jediného bitu zodpovedajúcej sedmice. Vo všeobecnosti môžeme povedať, že pomocou 2^k-1 pôvodných bitov nosiča vieme vložiť k informačných bitov. Na to používame Hammingov kód charakteristiky $(2^k-1, 2^k-k-1)$, kde $k \geq 1$ [87].

Definícia 5. *Priemerný počet zmien v bloku o veľkosti* 2^k-1 *je daný vzťahom*

$$h_k = \frac{1}{2^k}.$$

Definícia 6. *Saturáciu bloku o veľkosti* $2^k - 1$ *bitov definujeme vzťahom*

$$s_k = \frac{k}{2^k - 1}.$$

Definícia 7. *Efektivitou systému chápeme priemerný počet bitov správy, ktoré spôsobia zmenu jedného bitu v bloku originálneho nosiča o veľkosti* $2^k - 1$ *bitov. Určíme ju na základe vzťahu*

$$e_k = \frac{s_k}{h_k} = \frac{k}{1 - 2^{-k}}$$

Pri vkladaní informácie do nosiča je nutné pre dosiahnutie najvyššej miery efektivity maximalizovať hodnotu parametra k. Ten môžeme určiť na základe kapacity média a veľkosti vkladanej správy.

V našom steganografickom systéme vkladáme správu do najmenej významných bitov DCT koeficientov. Je zrejmé, že na účely vkladania nie je možné použiť koeficienty s hodnotou 0. Koeficienty s touto hodnotou sú sústredené v pravej dolnej časti matice, ktorá vznikne po kvantizovaní výstupnej matice diskrétnej kosínusovej transformácie. Bolo by podozrivé využívať tieto koeficienty na vkladanie správy, pretože v tejto oblasti matice sa nikdy nenachádzajú nenulové koeficienty.[3]

Keďže pri použití LSB techniky sa formujú tzv. *vzájomne previazané páry* hodnôt, nemožno na vkladanie infomácie použiť ani párovú hodnotu k hodnote 0, ktorou je 1. Preto z celkového počtu použiteľných koeficientov musíme odpočítať nielen nulové, ale aj jednotkové koeficienty.

Nakoniec nepoužívame ani absolútny člen (DC koeficient) žiadneho bloku údajov. Koeficient nesúci hodnotu amplitúdy je veľmi citlivý na akúkoľvek zmenu hodnoty, a porovnávaním hodnôt susedných blokov je možné odhaliť korelácie energie, ktoré by prípadná modifikácia tohto koeficientu mohla narušiť.

Výsledný vzťah pre výpočet kapacity (v súlade s definíciou na str. 9) nášho steganografického systému je

$$cap = C_{all} - C_{dc} - C_0 - C_1,$$

kde C_{all} je súčet všetkých koeficientov v médiu, C_{dc} je počet koeficientov reprezentujúcich amplitúdu signálu bloku, C_0 je počet nulových a C_1 počet jednotkových koeficientov.

Ak by sme chceli pomocou takéhoto steganografického systému preniesť správu o veľkosti 19 248 bitov (2156 bajtov) pomocou nosiča s kapacitou 258 696 bitov (32 337 bajtov), vypočítame saturáciu nosiča našou správou:

$$s_k = \frac{19\,238}{258\,696} \doteq 0.074,$$

čo činí približne 7,4%-né využitie kapacity nosiča. Táto hodnota je menšia než hodnota saturácie pre $k = 6$ ($s_6 \doteq 9.52\%$) a väčšia než hodnota pre $k = 7$ ($s_7 \doteq 5.51\%$). Preto na kódovanie takto veľkej správy pri danej kapacite môžeme využiť Hammingov kód s parametrom maximálne $k = 6$.

[3] to je vlastne dôvodom použitia transformácie z priestorovej do frekvenčnej oblasti — sústredenie signálu na jednom mieste

6.3 Stegoanalýza navrhnutého systému

Útoky na DCT koeficienty možno principiálne rozdeliť na:

- útoky bez znalostí štatistických zmien DCT koeficientov (necielená, tzv. *slepá* stegoanalýza),
- útoky so znalosťami štatistických zmien DCT koeficientov (*cielená* stegoanalýza).

Pre účely stegoanalýzy nášho systému je vhodné v skratke zhrnúť najdôležitejšie fakty vkladania správy z hľadiska jej stegoanalýzy.

Definícia 8. *Množinu všetkých kvantizovaných koeficientov obrázka budeme označovať ako DCT(I) alebo len jednoducho DCT. Kvantizované DCT koeficienty budeme podľa pozície v bloku označovať ako* $DCT(i,j)$ *pre* $i,j \in \{1,\ldots,8\}$.

- Na vkladanie (šifrovanej!) správy sa používajú len LSB bity AC koeficientov ($DCT(i,j)$, pre $(i,j) \neq (0,0)$), ktorých hodnoty sú rôzne od $\{0,1\}$,
- šifrovaná správa $M = m_0||m_1||\ldots$ sa vkladá po blokoch tak, že sa vkladá vždy k bitov správy (v aplikácii je k závislé od pomeru veľkosti správy a nosiča) do (2^k-1)–bitového bloku LSB bitov DCT koeficientov nosiča,
- (2^k-1)–tice(bloky) DCT koeficientov použité na vkladanie správy sa získajú zo zoznamu prípustných DCT koeficientov nosiča ich náhodným spermutovaním,
- vkladanie k–bitového bloku m_i šifrovanej časti do zodpovedajúceho (2^k-1)–bitového bloku LSB bitov DCT koeficientov nosiča sa realizuje nasledovne:
  - pre $k=1$ sa priamo vkladá 1 bit správy do LSB bitu DCT koeficientu,
  - pre $k \geq 2$ sa pomocou súčinu $H_c.m_i$ kontrolnej matice H_c Hammingovho kódu a m_i zistí poradové číslo DCT koeficientu, ktorého LSB sa zmení. V prípade, že m_i predstavuje nulový vektor správy, nemení sa žiadny bit zodpovedajúceho (2^k-1) bitového bloku.

Z pohľadu stegoanalýzy a štatistiky máme teda pre ľubovoľné k nasledovnú situáciu:

- pre konkrétny výber prípustných 2^k-1 DCT koeficientov je pravdepodobnosť zmeny nejakého LSB bitu DCT koeficientu bloku $p_{ch,blok} = 1-2^{-k} = \frac{2^k-1}{2^k}$;
- z matice Hammingovho kódu je zrejmé, že pravdepodobnosť $p_{ch,i}$ zmeny i–teho LSB bitu DCT bloku je pre všetky $i \in 1,\ldots,(2^k-1)$ rovnaká, a teda $p_{ch,i} = p_{ch}/(2^k-1) = 2^{-k}$;
- keďže blok (2^k-1) DCT koeficientov sa z množiny $DCT(I)$ vyberá náhodne, možno stanoviť pravdepodobnosť zmeny LSB bitu ľubovoľného DCT koeficientu ako:

$$p_{ch} = 2^{-k}.$$

Pri vkladaní teda pozorujeme zmenu ľubovoľného LSB bitu AC koeficientu s pravdepodobnosťou p^{-k}. Pozrime sa teraz na to, ako možno detegovať vkladanie s vyššie popísanou charakteristikou.

V nasledujúcom texte popíšeme všeobecné vlastnosti kvantizovaných DCT koeficientov bežných obrázkov z pohľadu stegoanalýzy a možností útokov na ne.

Vlastnosti DCT koeficientov čistých nosičov:

1. Hodnota kvantizovaných DCT koeficientov klesá — najväčšie hodnoty dosahujú DCT(i,j) koeficienty pre nízke hodnoty i, j. Pre väčšie hodnoty súradníc i, j v bloku sú typické nulové hodnoty DCT koeficientov.

2. DCT koeficienty bloku sú nekorelované — ľubovoľné dva DCT koeficienty bloku sú nezávislé a teda konkrétny DCT koeficient nemožno získať výpočtom zo zvyšných DCT koeficientov bloku. Obdobne sa vo všeobecnosti predpokladá,[4] že je obtiažne odvodiť hodnoty jednotlivých koeficientov bloku čistého nosiča z DCT koeficientov nosiča so správou. Táto vlastnosť „predurčuje“ DCT koeficienty ako vhodný nosič steganografickej informácie. Z tohoto dôvodu sa stegoanalýza `JPEG` obrázkov zameriava na jednotlivé DCT(i,j) koeficienty v rámci všetkých blokov a ich frekvencie (histogramy).

3. Histogramy jednotlivých AC DCT koeficientov majú podobný tvar — typický tvar histogramu možno vidieť na obrázku 6.4. Výnimku tvorí len histogram DC koeficientu DCT(0,0), ktorý sa aj z tohoto dôvodu nepoužíva na vkladanie správy. Ako vhodnú aproximačnú funkciu histogramov navrhuje Salle [69] vziať diskretizované zovšeobecnené Cauchyho rozdelenie pravdepodobnosti.

Vyššie uvedené vlastnosti priamo ovplyvňujú celú stegoanalýzu `JPEG` obrázkov nasledovným spôsobom:

- Z prvej vlastnosti plynie, že vloženie správy do nulových DCT koeficientov nosiča je ľahko detegovateľné. Preto aj väčšina steganografických metód (aj naša, ako sme už spomínali) modifikuje len nenulové DCT koeficienty.

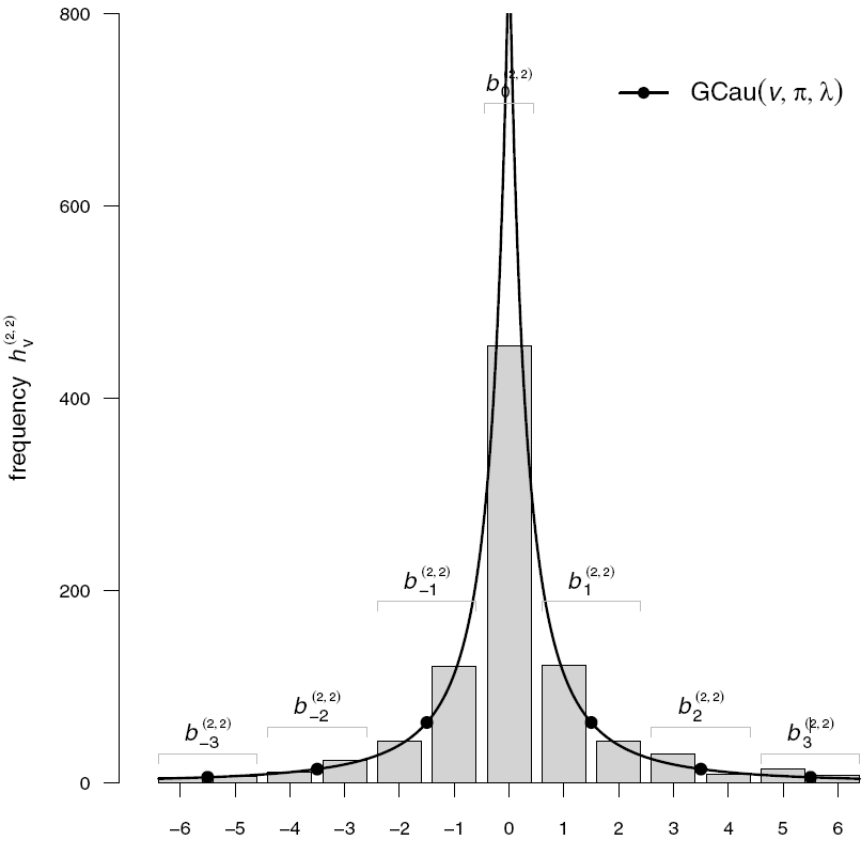
- Druhá vlastnosť hovorí o tom, že na JPEG steganografiu možno fakticky útočiť len cez histogramy jednotlivých DCT koeficientov.

  Pri útokoch na histogramy možno použiť štatistiky DCT koeficientov, ktoré sa obvykle delia na štatistiky:

  prvého rádu — pri týchto štatistikách sa berú do úvahy veličiny, ktoré popisujú údaje z hľadiska závislostí medzi jednotlivými vzorkami. Inými slovami, štatistiky prvého rádu sú invariantné voči permutáciám vzoriek. Táto vlastnosť je vlastná histogramom DCT koeficientov a všetkých ďalších veličín, ktoré z nich možno odvodiť (momenty rôznych stupňov).

  vyšších rádov — pri týchto štatistikách sa berú do úvahy vzťahy medzi vzorkami, resp. berie sa taktiež do úvahy ich pozícia (vo vektore dát). Toto napríklad zahŕňa aj mieru korelácie medzi susednými pixelmi pri vkladaní do LSB bitov nekomprimovaných obrázkov.

[4] zatiaľ nevieme o tom, že by to bolo dokázané



Obr. 6.4: Histogram pre koeficient DCT(2,2) [69].

Z pohľadu toho, aké štatistiky sa pri útokoch použijú, podobne voláme aj útoky — útok pomocou štatistík prvého rádu alebo vyšších rádov.

- Tretia vlastnosť hovorí o tom, že histogramy jednotlivých DCT koeficientov (okrem DCT(0,0)) majú všetky tvar podobný diskretizovanému Cauchymu rozdeleniu pravdepodobnosti. Teda detegovať steganografický algoritmus možno na základe zmeny tvaru histogramu.

Teraz bližšie popíšeme útoky na `JPEG` formát, ktoré možno využiť pri stegoanalýze nášho systému. Do úvahy prichádzajú nasledovné metódy stegoanalýzy:

- stegoanalýza algoritmu MB1 — útok pomocou štatistiky prvého rádu a
- kalibrácia JPEG histogramu — útok pomocou štatistiky vyšších rádov.

6.4 Kalibrácia JPEG histogramu

Metóda kalibrácie JPEG histogramu [31] bola pôvodne navrhnutá na cielenú stegoanalýzu algoritmu `F5`. S postupom času sa však stala základom pre väčšinu stegoanalytických metód `JPEG` formátu.

Cieľom kalibrácie je odhadnúť štatistické vlastnosti (histogramov jednotlivých DCT(i,j) koeficientov) pôvodného nosiča využitím *desynchronizácie* blokovej štruktúry v priestorovej oblasti.

Základná myšlienka detekcie steganografického algoritmu použitím kalibrácie JPEG histogramu spočíva v tom, že posunutie blokov rozmeru 8×8 nosiča, z ktorých sa počítajú DCT koeficienty, nespôsobuje významnú zmenu štatistických vlastností DCT koeficientov. Ak teda nastane pri desynchronizácii blokovej štruktúry významná zmena štatistických vlastností, možno hovoriť o úspešnej detekcii použitia steganografického algoritmu.

Pri praktickej realizácii sa postupuje nasledovne:

1. obrázok sa pomocou štandarnej JPEG dekomprimácie transformuje z frekvenčnej reprezentácie (reprezentácia pomocu DCT koeficientov) do priestorovej reprezentácie (farebné obrazové body)

2. v ďalšom kroku sa desynchronizuje bloková štruktúra — nové bloky pre výpočet DCT koeficientov sa vytvoria tak, že sa z obrázka „odstrihnú“ dva úzke okraje (jeden v horizontálnom a druhý vo vertikálnom smere), ktorých šírka je menšia ako 8 obrazových bodov. Následne sa štandardným spôsobom vytvoria bloky 8×8 pre výpočet DCT koeficientov

3. na výsledný obrázok sa opätovne aplikuje JPEG komprimácia s rovnakou kvantizačnou maticou, ako používal pôvodný obrázok.

Pri následnej analýze sa porovnávajú histogramy rovnakých DCT koeficientov v pôvodnom nosiči a desychronizovanom nosiči. Ako uvádza autor kalibračnej metódy, pri kalibrácii obrázka so stegosprávou sú výsledkom desynchronizovaného obrázka DCT koeficienty, ktorých štatistické vlastnosti (histogram DCT(i,j) koeficientov) sú oveľa bližšie k čistému nosiču než DCT koeficienty nekalibrovaného nosiča (obsahujúceho stegosprávu).

Autor uvádza nasledovný obrázok ako potvrdenie schopnosti kalibrácie histogramu detegovať steganografickú aplikáciu využívajúcu vkladanie do DCT koeficentov.

Ako možno vidieť z obrázkov 6.5 a 6.6, kalibráciou je možné získať pre vybrané DCT koeficienty pôvodné hodnoty s vysokou presnosťou. Preto výpočet rozdielu nekalibrovaného histogramu a kalibrovaného histogramu možno považovať za veľmi dobrý hrubý detektor vloženej stegosprávy. Pre zvýšenie presnosti detekcie stegosprávy je možné merať celkovú mieru odlišností všetkých 63 AC DCT koeficientov.

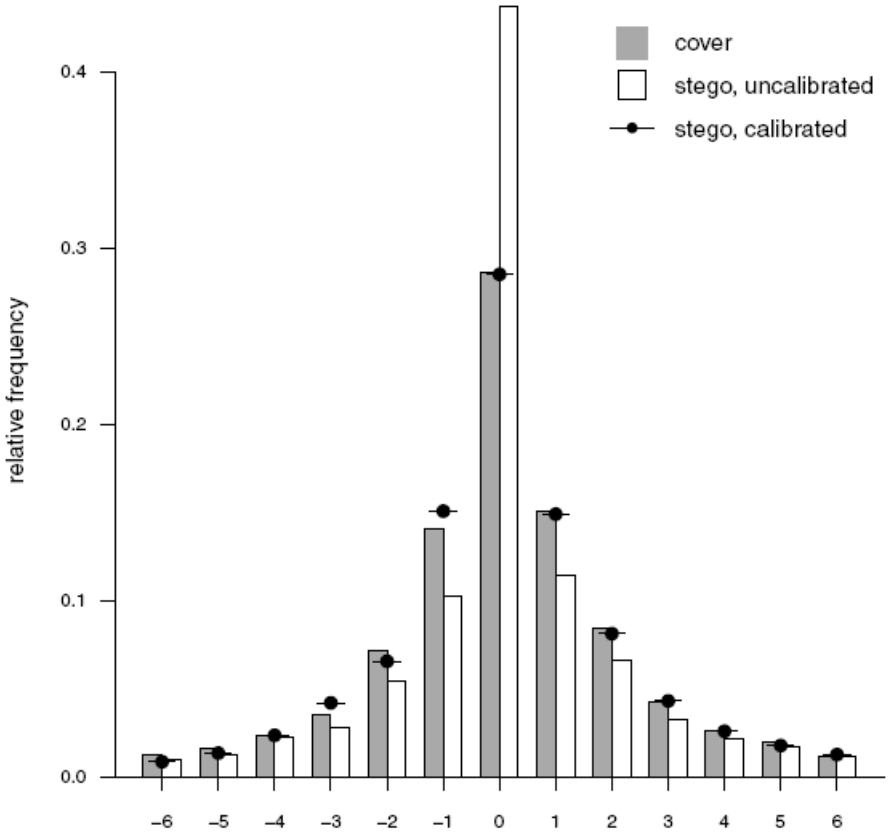
Poznámka 8. *Pri vkladaní správ do DCT koeficientov obrázka, ktorého histogramy sú znázornené v obrázkoch 6.5 a 6.6, bol použitý algoritmus F*5.

6.5 Algoritmus MB1 a jeho detekcia

Algoritmus MB1 vychádza z jednotného tvaru histogramov všetkých AC DCT koeficientov. Algoritmus je navrhnutý tak, aby sa po vložení správy do DCT koeficientov nezmenil „tvar“ histogramu, a tak bol odolný voči útokom využívajúcim štatistiky prvého rádu.

Rozoberme si pre názornosť, čo sa deje pri bežnom „náhodnom“ vkladaní[5] do LSB bitov DCT koeficientov. Pri vkladaní do LSB bitov nenulových DCT(i,j) koeficientov (i,j sú fixné)

[5] vzťahuje sa to aj na náš steganografický systém



Obr. 6.5: Histogram pre koeficient DCT(3,1) [64].

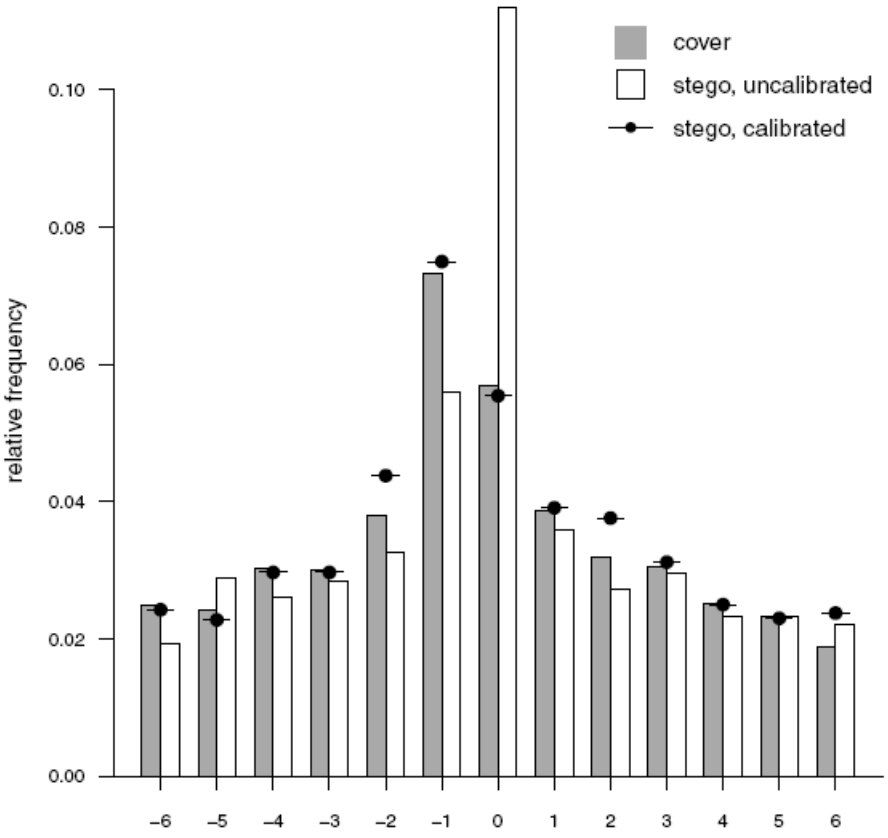
vzniknú páry hodnôt DCT koeficientov, ktoré sa menia jeden na druhý. Bez ujmy na všeobecnosti analyzujme histogram koeficientov $DCT(2,2)$. Nech h_u označuje počet DCT(2,2) koeficientov obrázka (stĺpec histogramu pre koeficient DCT(2,2)), ktoré majú hodnotu u. Nech sa pri „náhodnom“ vkladaní do LSB bitu DCT koeficientov zmení LSB bit s pravdepodobnosťou p_{ch}. To znamená, že pre páry $0 \leftrightarrow 1, 2 \leftrightarrow 3$ alebo všeobecnejšie $2u \leftrightarrow 2u+1$ sa zmení hodnota $2u$ na hodnotu $2u+1$ s pravdepodobnosťou p_{ch} a naopak $2u+1$ na $2u$ s rovnakou pravdepodobnosťou.

Nové hodnoty h'_{2u}, h'_{2u+1} histogramu teda možno odvodiť ako:

$$h'_{2u} = h_{2u} - p_{ch}h_{2u} + p_{ch}h_{2u+1},$$
$$h'_{2u+1} = h_{2u+1} + p_{ch}h_{2u} - p_{ch}h_{2u+1}.$$

Z uvedeného je zrejmé, že rozdiel nových hodnôt $h'_{2u} - h'_{2u+1}$ je závislý od rozdielu starých hodnôt $h_{2u} - h_{2u+1}$ podľa vzťahu

$$h'_{2u} - h'_{2u+1} = h_{2u} - h_{2u+1} + 2p_{ch}(h_{2u} - h_{2u+1}).$$



Obr. 6.6: Histogram pre koeficient DCT(1,2) [64].

Teda nové hodnoty h_{2u}, h_{2u+1} sa k sebe priblížia, čo má za následok deformáciu tvaru histogramu. Práve táto deformácia sa využíva pri útokoch s využitím štatistík prvého rádu.

Ako sme už vyššie spomenuli, algoritmus vkladania `MB1` bol skonštruovaný tak, aby zachoval „tvar“ histogramu. Vysvetlime si teraz podrobnejšie, ako mení algoritmus `MB1` štatistické vlastnosti DCT koeficientov.

Podobne ako pri našom steganografickom systéme, aj pri `MB1` sa na vkladanie používajú len nenulové koeficienty. Z hľadiska vkladania máme teda dvojice $2u+1, 2u$ pre $u > 0$ a $2u-1, 2u$ pre $u < 0$ kde $u \in Z$, ktoré sa menia jeden na druhý. Nech teda $h_v(i,j)$ predstavuje počet kvantizovaných DCT(i,j) koeficientov (i,j sú fixné) obrázka s hodnotou $v \in Z$. Teda hodnoty $h_v(i,j)$ určujú celý histogram koeficientu DCT(i,j). V literatúre sa zvykne táto hodnota nazývať v–ty „kôš“ (angl. *high precision histogram bin*). Je to vlastne súbor DCT koeficientov, ktoré majú konkrétnu hodnotu v. My budeme vzhľadom na neexistenciu slovenského ekvivalentu označovať tento kôš HPHB. V kontraste s tým sa spomína tiež u–ty kôš (angl. *low precision histogram bin*) $b_u(i,j)$, ktorý budeme označovať LPHB. Tento spravidla obsahuje práve dva (môže aj viac) HPHB. LPHB „kôš“ je tvorený všetkými pármi hodnôt, ktoré sa transformujú jeden na druhý.

Algoritmus `MB1` mení AC DCT koeficienty tak, že nová hodnota koeficientu nevypadne z pôvodného LPHB.

Salle v článku [69] navrhol využiť ako model pre $h(i,j)$ diskretizované zovšeobecnené Cauchyho rozdelenie pravdepodobnosti. Pri snahe o čo najpresnejšiu aproximáciu DCT histogramu Cauchyho rozdelením je snaha nájsť také parametre rozdelenia, ktoré najlepšie popisujú histogram pre DCT(i,j) koeficient.

Možno však ukázať, že tento model je prílišným zjednodušením skutočných nosičov, a teda použitím `MB1` algoritmu získame neprirodzene homogénne histogramy kvantizovaných DCT koeficientov. A práve táto neprirodzenosť sa dá použiť na detegovanie algoritmu `MB1`.

Pri detekcii algoritmu `MB1` sa vychádza z faktu, že hoci Cauchyho rozdelenie popisuje histogramy AC DCT koeficientov vcelku uspokojivo, pre prirodzené obrázky existujú „koše“ HPHB, ktoré sa od tohoto rozdelenia odchyľujú. Pri vkladaní pomocou algoritmu `MB1` sa tieto nevyhovujúce HPHB prispôsobujú modelu (tvaru Cauchyho rozdelenia). Preto model, ktorý berie do úvahy aj tieto odchýlky, tvorí z hľadiska prirodzených obrázkov lepší model. Využitie tohoto realistickejšieho modelu môže byť použité pri detekcii algoritmu `MB1`.

Detekciu `MB1` možno realizovať v nasledovných dvoch krokoch:

1. Aplikácia testu, ktorý rozlišuje modelu nevyhovujúce HPHB od vyhovujúcich. Tento test sa aplikuje nezávisle na všetky HPHB s nízkou presnosťou.
2. Počet kladných výsledkov (nevyhovujúcich HPHB) je porovnaný s empirickou hodnotou pre bežné obrázky.

Ak obrázok obsahuje menší počet modelu nevyhovujúcich HPHB, ako má zvyčajne čistý obrázok, je pravdepodobné, že histogram bol prispôsobený modelu použitím `MB1` algoritmu. Na obrázku č. 6.7 možno vidieť histogramy modelu, čistého nosiča a nosiča so stegosprávou.

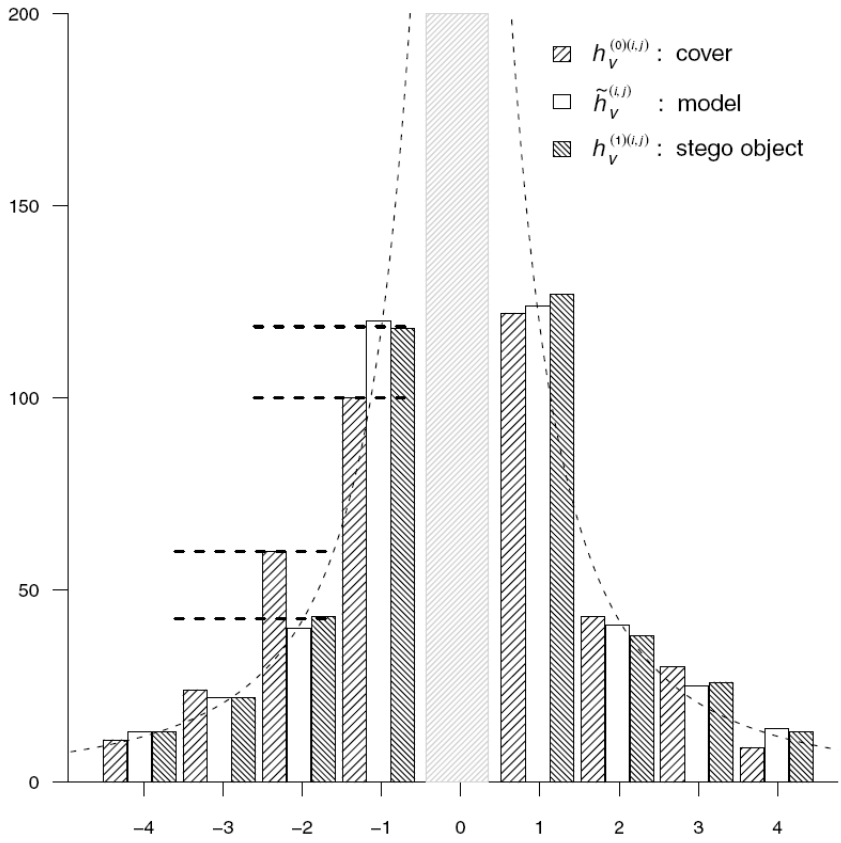
Ako možno z obrázka 6.7 vidieť, histogram nosiča so stegosprávou je blízky histogramu modelu vo všetkých HPHB. Rovnako tak možno vidieť odchýlky medzi histogramom čistého nosiča a histogramom modelu. Tento nesúlad možno merať využitím χ^2 testu. Pri ňom meriame odchýlku medzi pozorovanými frekvenciami košov s nízkou frekvenciou a očakávanými frekvenciami (z modelu). Dôvod využitia košov s nízkou frekvenciou je ten, že algoritmus `MB1` nemení ich hodnoty.

Testovanie, či konkrétny HPHB vyhovuje modelu, možno urobiť využitím kritickej hodnoty χ^{krit}. Toto tvorí test nulovej hypotézy toho, či pozorované a očakávané HPHB vychádzajú z rovnakého rozdelenia. Tento test odmieta nulovú hypotézu pre nevyhovujúce HPHB, keď hodnota štatistiky χ^2 je väčšia ako χ^{krit}.

Pre odhad počtu nevyhovujúcich HPHB v obrázkoch `JPEG` bolo testovaných 100 obrázkov z hľadiska HPHB s hodnotami $b_1(i,j), b_{-1}(i,j)$. Testovalo sa teda všetkých 63 AC DCT koeficientov na počet koeficientov s hodnotou 1 a -1. Dôvod, prečo sa vybrali DCT koeficienty s nízkou hodnotou, je zrejme ten, že pri nich možno očakávať najväčšie merateľné odchýlky po vložení stegosprávy.

Z hľadiska realizovaného χ^2 testovania HPHB je ešte vhodné spomenúť dva dôležité fakty:

1. Test je nespoľahlivý pre malé čísla (zamieta modelu nevyhovujúce HPHB). Z tohoto dôvodu neboli brané do úvahy HPHB s frekvenciami menšími ako 3.



Obr. 6.7: histogramy HPHB DCT koeficentov [64].

2. Spoľahlivosť testu závisí na počte nenulových DCT koeficientov. Toto číslo sa mení spolu s rozlíšením obrázka a kvantizačným faktorom odvodeným z kvality q snímky. Ako uvádzajú autori, je vhodné nastaviť parametre na hodnoty $\chi^2_{krit} = 4$ a $q = 0,8$.

Z predchádzajúceho textu teda vyplýva, že pri stegoanalýze nášho steganografického systému je možné použiť nasledovné stegoanalytické metódy:

- kalibrácia `JPEG` histogramu,
- detekcia algoritmu `MB1` upravená pre naše účely — t.j. zistenie počtu HPBH bitov a ich porovnanie s empirickými hodnotami,
- výskumná metóda — t.j. testovanie zhody tvaru histogramov pre jednotlivé DCT koeficienty a ich zhoda s empirickými hodnotami bežných obrázkov.

Kapitola 7

Steganografický systém založený na komprimovanom videu

Poslednou oblasťou návrhu steganografických systémov, ktorej sme sa venovali počas obdobia výskumu, bola konštrukcia steganografického systému založeného na komprimovanom videu. Hlavným cieľom nášho výskumu bolo zistiť možnosti implementácie steganografického systému, ktorý by nebol závislý od použitého kodéra videa. Preto sme sa zamerali na štruktúru nosičov pohyblivého obrazu, a z nich sme vybrali kontajner `mp4`.

Jeho výhodou je široká podpora zariadení a softvéru. Súborový formát `avi` je na ústupe, a jeho miesto preberá formát `mp4`. Okrem toho patrí medzi najpoužívanejšie formáty na internetových úložištiach multimediálnych údajov. To je dôležité, pretože návrh steganografického systému by mal zohľadňovať aj transport správy, ktorá je viazaná na nosič. V prípade veľkého množstva údajov je nutné hľadať vhodné prenosové kanály. Najvhodnejším (a najmenej podozrivým) je prenos údajov pomocou Internetu.

V tejto kapitole predstavíme dve nové techniky vkladania informácií do `mp4` súborov:

1. prvá z nich je založená na modifikovaní počtu snímok v tzv. štruktúre GOP (z angl. *Group of Pictures*), ktorú predstavíme ďalej v texte, a
2. druhá využíva na kódovanie informačných bitov počty snímok pri striedajúcich sa údajových prúdoch v ich fyzickom uložení v `mp4` kontajneri.

Hoci hlavnou požiadavkou, ktorú sme si pri tejto štúdii stanovili, je nezávislosť od použitého video kodéra, predsa, ako ukážeme neskôr, vhodnosť videa na použitie tohto typu steganografie veľmi silno ovplyvňujú nastavenia kódovania a vlastnosti snímanej scény.

Štruktúra tejto kapitoly je nasledovná: najprv v časti 7.1 objasňujeme základné entity využívané v navrhnutom steganografickom systéme. V nasledujúcej časti definujeme požiadavky kladené na náš systém. Návrh steganografického systému opisujeme v časti 7.3. V časti 7.4 zhŕňame výsledky niektorých meraní zameraných na vhodnosť použitia navrhnutého steganografického systému. Na záver diskutujeme obmedzenia implementácie tohto systému.

7.1 Štruktúry použité v návrhu steganografického systému

Navrhnutý steganografický systém využíva dve štruktúry, ktoré sú prítomné v `mp4` súboroch, a zároveň pri manipulácii s nimi nie je nutné poznať implementáciu kodéra videa. Jedná sa o pravidelne sa opakujúce zoskupenie snímok (GOP) a striedanie video a audio prúdov.

Štruktúra GOP

GOP označuje skupinu obrázkov s presne definovanou štruktúrou. Informácie o tejto štruktúre spolu s ďalšímu súvislosťami boli čerpané z [63]. Štandard `MPEG-1` zaviedol niekoľko typov snímok: I, B a P.[1]

Grafická informácia v I snímkach je prenášaná pomocou stratovej kompresie, ktorá vychádza zo štandardu JPEG. Jedná sa o tzv. *vnútro–snímkové* kódovanie, angl. *intra–coding*, od čoho pochádza aj označenie tohto typu snímok. I snímky obsahujú plnú grafickú informáciu, a na rozdiel od ostatných typov nepotrebujú na dekódovanie žiaden iný.

I snímky majú medzi ostatnými špeciálne postavenie. Definujú začiatok skupiny obrázkov (GOP). Každý GOP je určený svojou počiatočnou I snímkou a ostatnými snímkami (typu P alebo B) až do nasledujúcej I snímky, ktorá k tomuto GOP už nepatrí, ale určuje začiatok nasledujúceho GOP. Obrázok č. 7.1 zobrazuje zapúzdrenie GOP do video prúdu. Celý video prúd je zložený zo skupín snímok. GOP je tvorený skupinou snímok. Každá snímka sa skladá z tzv. *behov makroblokov*, a každý makroblok sa skladá zo štvorice matíc rozmerov 8×8.

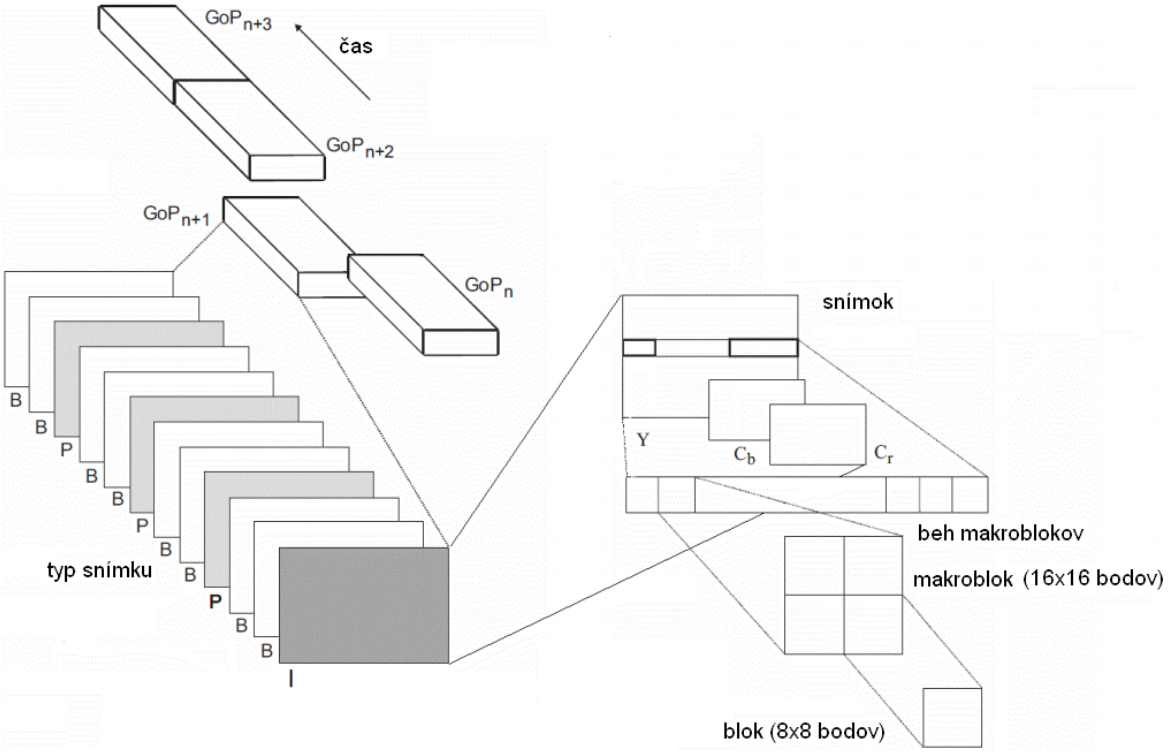
Na rozdiel od I snímok sú obrázky typu P a B kódované *mimo–snímkovo* (angl. *outer–coding*). Neobsahujú plnú grafickú informáciu, ale iba odkazy na časti predošlých alebo nasledujúcich snímok.

Každá P snímka vzniká na základe predošlej I alebo P snímky. Jedná sa o tzv. *doprednú* predikciu, nakoľko dekódované snímky určujú snímku v budúcnosti (vzhľadom na postupnosť dekódovania snímok). Označenie tohto typu snímok pochádza z anglického *prediction* zo štandardu H.261. Až `MPEG-1` zaviedol B snímky, takže bolo potrebné rozlišovať, o aký typ predikcie ide.

Označenie B snímok teda pochádza z anglického *backward*, ktoré naznačuje, že sa jedná o spätnú predikciu. Teda aktuálne dekódovaný obrázok sa môže za istých okolností stať vzorom pre rekonštrukciu snímky v minulosti (vzhľadom na časovú postupnosť dekódovania snímok). Hoci názov môže evokovať, že tvorba B snímok je obmedzená na spätnú predikciu, v skutočnosti môžu B snímky vznikať aj z doprednej predikcie. Je na kodéri, aby rozhodol, s akou váhou sa zoberie spätná a s akou dopredná predikcia, a ako sa určí výsledný obrázok. Väčšinou sa pre B snímky používa obojsmerná interpolačná predikcia. Najprv sa určí dopredná, potom spätná predikcia. Kodér na základe rôznych obmedzení (predovšetkým veľkosť chyby po dekódovaní oproti originálu) rozhodne o váhovaní jednotlivých predikcií.

Zásadným obmedzením B snímok je skutočnosť, že vnášajú oveľa väčšiu chybu do obrazu než P snímky. Preto štandardy rodiny `MPEG` neumožňujú využívať B snímky ako zdroj ďalšej predikcie. Jedná sa o finálne snímky, na ktoré sa nemôžu odkazovať žiadne ďalšie ani predošlé snímky video sekvencie.

[1] I a P snímky boli síce používané už štandardom H.261, ale až od štandardu MPEG–1 sa im dostalo viac pozornosti



Obr. 7.1: Zapúzdrenie GOP–ov do video prúdu v štandarde `MPEG-1` [63].

Počet použitých P alebo B snímok v GOP–e nie je štandardom `MPEG` nijako obmedzený. V štandarde sa dĺžka GOP–u udáva parametrom N. Vzdialenosť medzi P a I snímkami zvykne byť označovaná parametrom M. Ukážku GOP–u s parametrami $N = 12$ a $M = 3$ možno vidieť na obrázku č. 3.1 na strane 29.

Keďže dĺžka GOP–u môže byť variabilná, a ona taká v skutočnosti aj je[2], v našom steganografickom systéme využijeme túto vlastnosť na ukrytie informácie.

Striedanie údajových prúdov v `mp4` kontajneri

Súborový formát `mp4` je navrhnutý na uchovávanie rôznych údajových prúdov. Medzi najpoužívanejšie patria audio a video prúdy. Samotný kontajner (formát súboru) je podrobne špecifikovaný medzinárodným štandardom ISO 14496-14.

Formát je odvodený od kontajneru `QuickTime`, ktorý vyvinula firma Apple. Fyzickú organizáciu údajov (relevantných pre video prúd) znázorňuje obrázok č. 7.2. Všetky údaje sú organizované v tzv. *atómoch*. Na začiatku každého atómu sa nachádza jeho identifikátor a dĺžka [28]. Atómy sa môžu vnárať, takže jeden atóm môže obsahovať viacero iných.

[2] predovšetkým kvôli nutnosti kodéra zavádzať I snímky vždy, keď chyba kódovania vzrastie nad akceptovanú hodnotu — a to sa deje napríklad pri prudkej zmene scény

Vzhľadom na náš steganografický systém sú zaujímavé atómy `moov`, `mdat`, `stbl`, `stts`, `stss`, `stsc`, `stsz` a `stco`:

`moov` — zapúzdruje celý kontajner, obsahuje kontrolné a riadiace údajové štruktúry;

`mdat` — uchováva údaje;

`stbl` — nachádzajú sa v ňom informačné tabuľky ohľadom jednotlivých dátových prúdoch;

`stts` — obsahuje časové údaje o dĺžkach jednotlivých dátových prúdoch;

`stss` — v tomto atóme sa nachádza zoznam snímok, ktoré môžu byť priamo sprístupnené; jedná sa vlastne o I snímky, pretože iba tie obsahujú úplnú grafickú informáciu, a nepotrebujú ďalšie snímky na svoje dekódovanie;

`stsc` — nachádza sa tu tabuľka, ktorá na každom riadku obsahuje informácie o tzv. *zhluku* (angl. *chunk*) údajov; zhlukom sa chápe sled viacerých snímok za sebou (nemusí sa jednať o úplnú GOP štruktúru) — podrobnejšie bude termín zhluku vysvetlený v nasledujúcom texte;

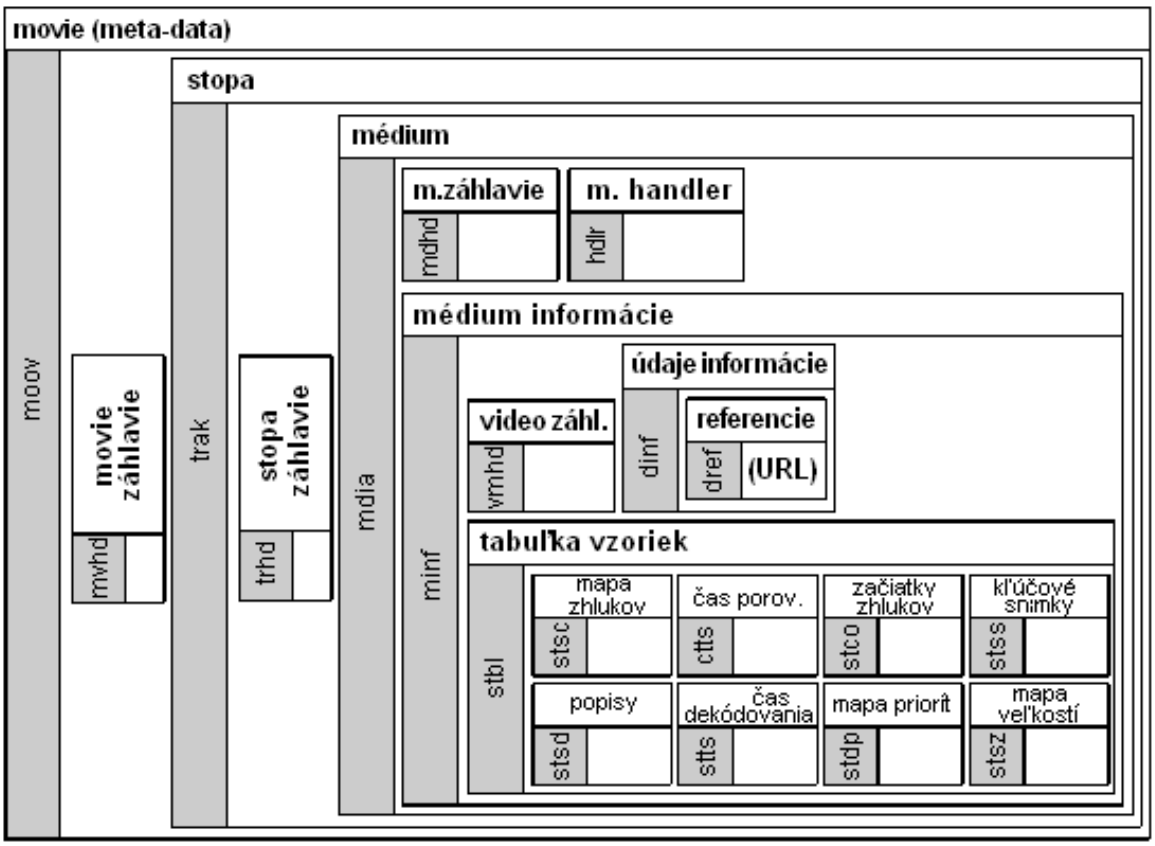
`stsz` — obsahuje odkazy na reálne začiatky jednotlivých snímok, ktoré sa nachádzajú v atóme `mdat`;

`stco` — udržiava zoznam začiatkov jednotlivých zhlukov údajov, ktoré sú popísané atómom `stsc`.

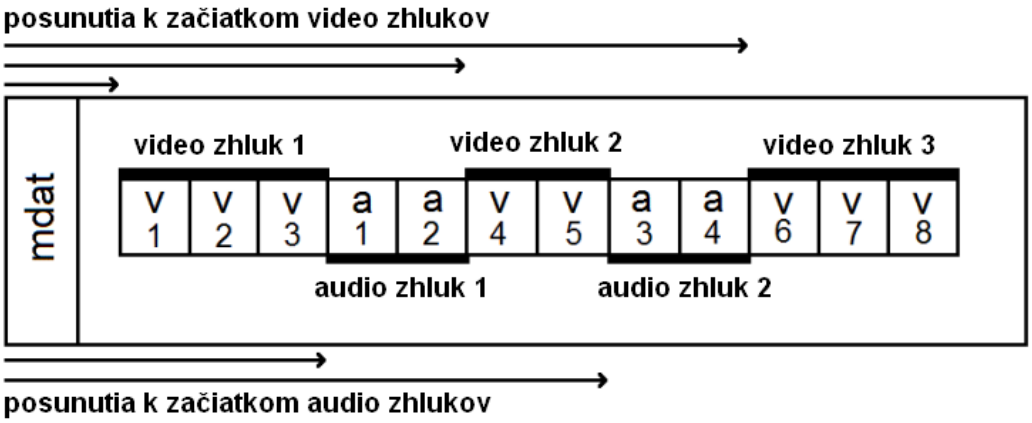
Atóm `mdat` obsahuje samotné bitové údajové prúdy, ktoré sa vzájomne striedajú. Kontainer `mp4` bol navrhnutý s ohľadom na prenos v reálnom čase aj pomocou nízkokapacitných komunikačných kanálov, preto pri ukladaní údajov využíva techniku časového multiplexu. Každému údajovému prúdu je vyhradený isté časové kvantum, a v rámci tohto má daný údajový prúd k dispozícii celú kapacitu pásma. Preto sa jednotlivé prúdy striedajú. Údaje, ktoré sa počas prideleného časového kvanta stihnú spracovať, tvoria už vyššie spomenuté *zhluky*. Ukážku striedania audio a video prúdov v `mdat` atóme znázorňuje obrázok č. 7.3. Ako vidno, zhluky môžu mať rozličné veľkosti a rôzny počet snímok.

7.2 Požiadavky kladené na steganografický systém

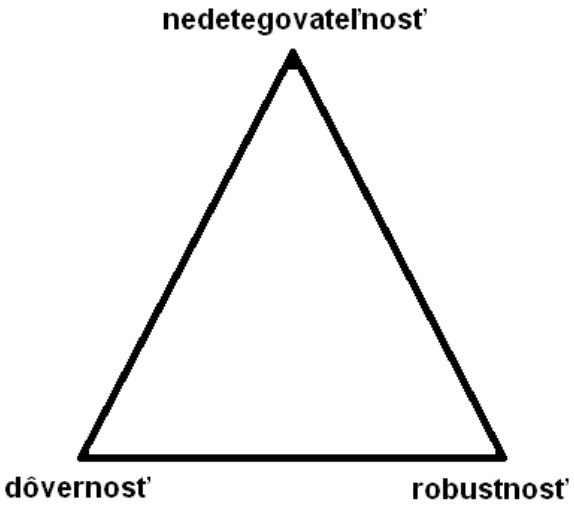
Návrh steganografických systémov sa snaží dosiahnuť bezpečnosť v zmysle troch základných vlastností: súkromie, robustnosť a nedetegovateľnosť [15]. Tieto vlastnosti nie je možné dosahovať v plnej miere súčasne, pretože ich požiadavky sú často nezlučiteľné (alebo dokonca protichodné). Môžeme ich reprezentovať vrcholmi trojuholníka (viď obrázok č. 7.4). Ak sa pokúsime zvýšiť úroveň bezpečnosti implementáciou jednej z vlastností, zvyšné dve musíme zanedbať. Preto sa často volí kompromis medzi implementáciou všetkých troch. V nasledujúcom texte objasníme, akým spôsobom chceme dosahovať bezpečnosť z pohľadu týchto troch vlastností v našom steganografickom systéme.



Obr. 7.2: Štruktúra atómov kontaineru `mp4`.



Obr. 7.3: Striedanie prúdov zvuku a videa v atóme `mdat` kontaineru `mp4` [43].



Obr. 7.4: Vzťahy medzi hlavnými cieľmi steganografickej bezpečnosti.

Dôvernosť

Dôvernosť chápeme ako schopnosť zachovať tajomstvo v tom zmysle, že neautorizovaná osoba nie je schopná získať (v reálnom čase) tajnú informáciu.

Návrh nášho steganografického systému zakladáme na Kerckhoffovom princípe: predpokladáme, že schému použitého steganografického systému útočník pozná. Skúsenosť z histórie ukazuje, že dôvernosť založená na utajovaní použitej transformačnej (alebo kódovacej) schémy nevedie k zvýšeniu bezpečnosti. Metódami reverzného inžinierstva v spolupráci so sociálnym inžinierstvom je vo väčšine prípadov možné rekonštruovať proces spracovania informácií utajenými technikami. Utajovanie schémy môže spôsobiť iba isté časové oneskorenie rekonštrukcie.

Preto nami navrhnutá schéma je verejná a na dosahovanie dôvernosti využívame predspracovanie vkladanej steganografickej informácie pomocou vhodného šifrovacieho algoritmu. Do steganografického média sa vkladá až zašifrovaný text. Útočník, ktorý pozná použitú steganografickú schému, musí pre narušenie staganografickej schémy[3] zamerať svoju pozornosť na narušenie použitého kryptografického systému.

Navyše, vhodné štatistické vlastnosti takto upravenej správy možno testovať vzhľadom na želané požiadavky systému (niektoré z týchto požiadaviek pre náš systém budú spomenuté ďalej). Napríklad dostatočnú náhodnosť možno testovať pomocou štandardizovaných alebo proprietárnych (viď napr. [34]) balíčkov štatistických testov.

[3] v prípade dôvernosti ide nielen o potvrdenie existencie správy v nosiči, ale aj o získanie samotnej vloženej informácie

Robustnosť

Robustnosť sa zvyčajne spája s transportom správy cez komunikačný kanál. Samotný transport možno z hľadiska procesov rozdeliť na:

1. poslanie správy,
2. prenos pomocou komunikačného kanála a
3. prijatie správy legitímnym príjemcom.

Robustnosť je meraná napríklad mierou úspešnosti prenosu správy (tzv. *prežitím*) v steganografickom nosiči počas transportu média komunikačným kanálom, pričom nesmie prísť k poškodeniu (ktoré by znemožnilo dekódovanie) vloženej steganografickej informácie. Preto je jedným z cieľom výskumu v oblasti steganografie vyvíjanie takých algoritmov na vkladanie správ do nosičov, ktoré zabezpečia integritu tajnej informácie aj v prípade modifikácie steganografického média počas prenosu komunikačným kanálom.

Nami navrhovaný steganografický systém používa na prenos informácie interné štruktúry videa kódovaného podľa štandardu `MPEG` a spôsob fyzického uloženia multimediálnych údajov v kontajneri `mp4`. Štruktúra GOP je nezávislá na použitom súborovom formáte, a teda nezávislá na fyzickej štruktúre údajov. Preto je tento steganografický systém odolný voči zmene súborového formátu uchovávajúceho samotné obrazové a zvukové informácie. Avšak voči zmene použitého kódovania nie je odolný. Nakoľko štandard nepredpisuje do úplných detailov štruktúru GOP, akýkoľvek kodér môže (a tak to aj robí) meniť štruktúru video prúdu podľa potrieb algoritmov, ktoré kontrolujú kvalitu výstupného kódovania. Zmena kódovania vynucuje aj zmenu fyzického usporiadania údajov v použitom multimediálnom kontajneri. Preto prekódovanie videa môže viesť k zničeniu steganografickej informácie vloženej pomocou fyzického uloženia video prúdu.

Nedetegovateľnosť

Samotná nedetegovateľnosť zahŕňa dve požiadavky:

1. prvou je nemožnosť rozhodnutia, či médium obsahuje skrytú správu, iba na základe štúdia zvolených charakteristík nosiča a
2. druhou je nemožnosť potvrdenia (alebo vyvrátenia) existencie skrytej správy v predpokladanom steganografickom médiu, aj keď útočník pozná algoritmus vkladania správy do nosiča.

Aby bola splnená prvá požiadavka nedetegovateľnosti, je nutné, aby steganografický algoritmus zohľadňoval tie štatistické vlastnosti nosiča, ktoré sa menia vkladaním správy. Navyše, druhá požiadavka vyžaduje, aby nebolo možné na základe znalosti algoritmu vkladania informácie do nosiča odlíšiť médium obsahujúce vloženú správu od čistého média.

Aby sme predišli detegovaniu vkladania správy na základe štúdia štatistických vlastností nosiča, v našom návrhu steganografického systému zohľadňujeme jeho štruktúru. Pred samotným vložením zašifrovanej správy sa vykonáva zmena kódovania, aby postupnosť vkladaných bitov korešpondovala s vlastnosťami binárnej postupnosti, ktorá je modifikovaná. Na efektívne doladenie výslednej postupnosti je vhodné použiť výstup optimalizačného modulu opisovaného v kapitole 8.

Pri prvej manipulácii s nosičom je potrebné overiť, či je možné použiť dané video v takom kódovaní, v akom sa nachádza. Vzhľadom na to, že na vkladanie informácie využívame nepravidelný počet snímok v GOP štruktúre, je nutné, aby bol tento počet variabilný a neperiodický. Ak táto podmienka nie je splnená, nemožno použiť navrhovaný steganografický systém bez možnosti vzbudenia podozrenia.

Pri použití štruktúry striedania video a audio prúdov na fyzickej vrstve toku dát je situácia podobná. Ak sa prúdy striedajú s pevne definovanou periódou, akýkoľvek zásah do tejto štruktúry môže vyvolávať podozrenie na prenos utajenej informácie. Preto je znovu nutné pred vkladaním informácie kontrolovať dostatočnú variabilitu uvedenej štruktúry a pri vkladaní informácie zohľadňovať vlastnosti zodpovedajúcej binárnej postupnosti.

7.3 Návrh steganografického systému

Využitie štruktúry GOP

Pripomíname, že návrh steganografického systému je založený na skutočnosti, že štandardy kodérov rodiny `MPEG` nešpecifikujú počet snímok ani v štruktúre GOP ani v zhlukoch jednotlivých prúdov v kontajneri `mp4`. Nami navrhovaná metóda spočíva v kódovaní vkladanej informácie pomocou počtu snímok v jednotlivých sledovaných štruktúrach. Predpokladom na úspešné vloženie informácie do súboru `mp4` je, ako bolo spomínané v predošlej časti, dostatočná variabilita počtov snímok v jednotlivých štruktúrach.

Kódovanie vkladanej informácie sa robí rovnakým spôsobom v oboch štruktúrach. V prípade GOP sa kóduje jeden bit informácie pomocou počtu P a B snímok v skupine. Ak je potrebné zmeniť tento počet, robí sa tak na konci celej skupiny, aby sa minimalizoval vplyv degradácie výsledného obrazu vložením alebo zmazaním P či B snímky. Chyba nastáva v dôsledku toho, že pri vkladaní sa kopíruje posledná snímka, nakoľko podľa požiadaviek kladených na steganografický systém nezasahujeme do kódovania videa.

Využitie štruktúry zhlukov

V prípade použitia štruktúry fyzického umiestnenia prúdu údajov zodpovedajúcich videu je situácia nasledovná.

Údajové prúdy sa v kontajneri `mp4` nachádzajú v atóme `mdat`. Sú uložené striedavo, v závislosti od použitého časového multiplexu, ktorý vyhradzuje každému z kanálov rovnaké

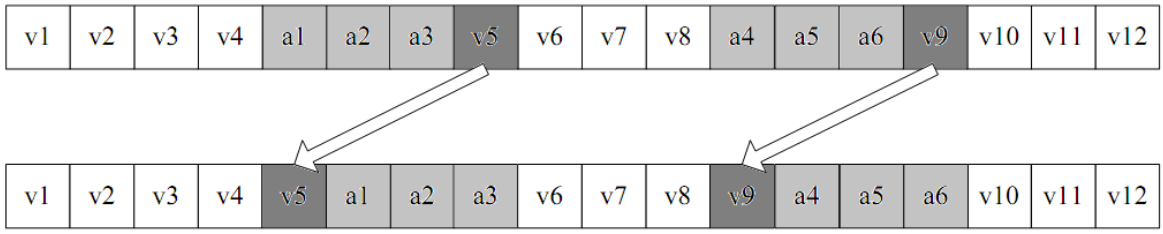
15	15	15	15	15	16	15	15	15	15	15	16	15	15	15	15	15	16	15	15	15	15	15	16	15	15	15
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0

Obr. 7.5: Príklad video zhlukov v kontajneri `mp4` [42].

časové kvantum. Teda výsledný počet rámcov videa uložených v zhlukoch závisí aj od špeciálneho parametra kodéra, ktorý určuje šírku bitového toku za sekundu (angl. *bitrate*).

Na ukážku uvádzame na obrázku č. 7.5 počty snímok v jednotlivých zhlukoch veľkosti 15 a 16. Ak je parameter bitrate pevne daný (statický), počty snímok v jednotlivých zhlukoch nie sú konštantné, pretože veľkosti jednotlivých snímok nie sú (vo všeobecnosti) rovnaké. Túto skutočnosť podrobnejšie objasníme v časti 7.4. Od parametra bitrate a želanej výslednej kvality videa (nepriamo) závisí aj nastavenie parametra, ktorý určuje počet snímok v toku za sekundu (tzv. *framerate*). Na spomenutom obrázku sa nachádza zodpovedajúci sled počtov snímok v jednotlivých zhlukoch pre jeden ukážkový framerate. Môžeme si všimnúť, že sa pravidelne striedajú dve hodnoty: 15 a 16. Pre iný video súbor to môžu byť iné dve hodnoty. Pri konkrétnom framerate stačí minimálne jedna a maximálne dve hodnoty na časovú synchronizáciu videa.

Túto skutočnosť môžeme využiť a kódovať pôvodné postupnosti dvoch čísel (resp. jedného čísla) pomocou binárnej postupnosti. Menšiemu číslu priradíme hodnotu 0 a väčšiemu 1. Prvý riadok na obrázku č. 7.5 zobrazuje počty snímok v jednotlivých zhlukoch, a v druhom riadku sa nachádza zodpovedajúca binárna postupnosť.



Obr. 7.6: Príklad kódovania informácie pomocou zhlukov [42].

Steganografický systém založený na zhlukoch kóduje informačné bity správy pomocou počtov snímok v jednotlivých zhlukoch. Je však dôležité si uvedomiť, že zmena fyzického usporiadania údajov v kontajneri `mp4` so sebou nesie nutnosť zmeny riadiacich a informačných štruktúr kontajnera. Tie sa nachádzajú v príslušných atómoch (viď obrázok č. 7.2): `stts`, `stss`, `stsc`, `stsz` a `stco`.

Napríklad (viď obrázok č. 7.6) ak chceme vložiť informačný bit 0, môžeme ho kódovať párnym počtom snímok v zhluku. Musíme teda skontrolovať počet snímok v príslušnom zhluku. Ak je tento počet párny, nič sa nemení. Ak je však nepárny, je potrebné zmeniť počet snímok v tomto zhluku. To je možné dosiahnuť odobratím snímky z predmetného zhluku a jej pridaním do nasledujúceho.

Ak vykonávame takého zásahy do fyzického usporiadania prúdov v kontajneri, musíme aktualizovať všetky relevantné informácie, ako ukazuje obrázok č. 7.6.

Podotýkame, že túto techniku vkladania je možné použiť nielen pre video prúd, ale aj pre akýkoľvek iný obsiahnutý v multimediálnom kontajneri `mp4`.

Algoritmus vkladania správy

Nakoľko funkcia kódovania informačných bitov správy je rovnaká pre obe štruktúry, kroky algoritmu objasníme iba na štruktúre GOP.

V situácii, keď v procese vkladania správy kódujeme informačný bit 0 párnou dĺžkou GOP a bit 1 nepárnou, môžu nastať nasledovné situácie:

1. Ak chceme vložiť bit 0:
   a) dĺžka GOP je párna: žiadna zmena,
   b) dĺžka GOP je nepárna: posledná snímka v GOP je odstránená a kvôli zachovaniu synchronizácie sa zdvojí posledná snímka nasledujúcej GOP.
2. Ak chceme vložiť bit 1:
   a) dĺžka GOP je nepárna: žiadna zmena,
   b) dĺžka GOP je párna: posledná snímka v GOP je odstránená a kvôli zachovaniu synchronizácie sa zdvojí posledná snímka nasledujúcej GOP.

Ako sme už skôr spomínali, zdvojenie poslednej snímky skupiny vnáša do videa chybu. Táto chyba môže byť kompenzovaná vložením tzv. prázdnej snímky, ktorá bude referencovať predošlú bezo zmeny.

Nasleduje algoritmus vloženia správy do video sekvencie:

Krok 1: Over vhodnosť video súboru (nosiča) vzhľadom na variabilitu dĺžok GOP a zhlukov použitých dátových prúdov. Zároveň sa v tomto kroku zistí počet použiteľných GOP.

Krok 2: Ak je variabilita menšia než (napríklad aj používateľom) stanovená hranica, algoritmus končí.

Krok 3: Zašifruj správu, ktorá sa má vložiť.

Krok 4: Over dostatočnosť kapacity média. Kapacita je daná počtom GOP–ov, ktoré môžu byť použité na vloženie informácie. Kapacitu je vo všeobecnosti možné iba odhadnúť, presne sa dá určiť iba pre konkrétne zvolenú správu a nosič. Je to dané skutočnosťou, že kódovaním bitu, ktorý mení dĺžku istej GOP, sa modifikuje aj nasledujúca GOP, čo môže mať vplyv na počet použiteľných GOP.

Krok 5: Ak je kapacita nosiča nedostatočná, koniec algoritmu.

Krok 6: Vlož informáciu do videa pomocou vyššie spomínanej kódovacej schémy (založenej buď na zmene dĺžky GOP alebo zmene dĺžky údajových zhlukov vo fyzickej reprezentácii jednotlivých prúdov).

Algoritmus dekódovania informácie:

Krok 1: Pre všetky použiteľné GOP vo video súbore dekóduj vložené informačné bity podľa počtu snímok nasledovne:

1. ak je dĺžka GOP párna, dekóduj bit 0,
2. ak je dĺžka GOP nepárna, dekóduj bit 1.

Krok 2: Dešifruj získanú správu.

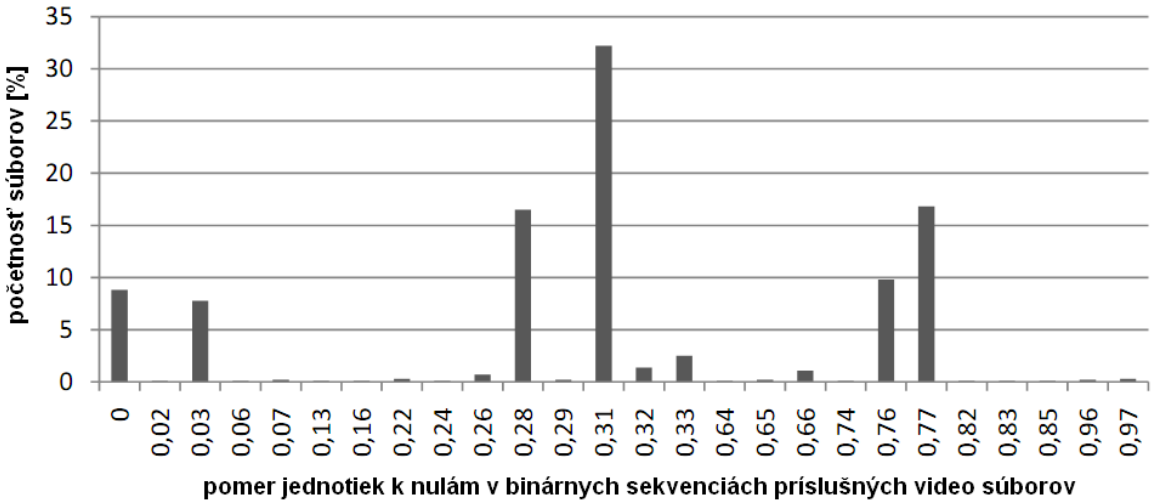
7.4 Merania

V časti 7.2 sme diskutovali medzi inými hlavnými cieľmi bezpečnosti steganografického systému aj nedetegovateľnosť. Predstavený steganografický systém používa špeciálnu fyzickú štruktúru zhlukov snímok na vkladanie informácie. Táto štruktúra, ako bolo skôr povedané, je úzko previazaná na parameter kodéra označovaný ako *framerate*. V súvislosti s týmto parametrom sme vyšetrovali 1000 videí pochádzajúcich z internetového multimediálneho úložišťa *YouTuBe*. Všetky videá pochádzajúce z neho majú taký framerate, že každému video súboru na tejto službe zodpovedajú maximálne dve hodnoty dĺžok video zhlukov. Analyzovali sme tieto dĺžky, a výsledok možno vidieť na obrázku č. 7.7.

613 video súborov malo jednu z dĺžok video zhlukov 15, 232 malo základnú dĺžku 13 a 111 malo 16. 95% všetkých video súborov malo dĺžky video zhlukov 13, 15 alebo 16.

Obr. 7.7: Video súbory pochádzajúce z YouTube klasifikované na základe dĺžok video zhlukov.

Následne sme analyzovali charakteristiku binárnych postupností, ktoré zodpovedali jednotlivým zhlukom video súborov. Základným poznatkom je skutočnosť, že žiadna postupnosť nevykazuje charakteristiku rovnomerného rozdelenia (núl a jednotiek). Histogram početností súborov, ktorých pomer núl k jednotkám v postupnosti bol rovnaký, sa nachádza na obrázku č. 7.8.



Obr. 7.8: Rozdelenie video súborov vzhľadom na pomer núl v zodpovedajúcich binárnych postupnostiach.

Binárne postupnosti vytvorené z každého skúmaného video súboru sa dajú rozdeliť do 33 rôznych tried (viď obrázok č. 7.9). Vzorka, ktorá korešponduje každej triede, sa v príslušných súboroch periodicky opakuje od začiatku až do konca. To je spôsobené výberom hodnoty parametra framerate. Je zrejmé, že tieto vzorky nie sú náhodné. Preto nie je nijaká možnosť využitia tejto štruktúry na vkladanie steganografickej informácie bez vzbudenia podozrenia. Ak sa nejaká správa vloží, binárna sekvencia prislúchajúca danému videu sa zmení, a (s vysokou pravdepodobnosťou) nebude korešpondovať žiadnej vzorke z jednotlivých tried zobrazených na obrázku č. 7.9). Útočníkovi teda stačí sledovať iba túto charakteristiku, aby odhalil existenciu skrytej informácie.

Aby sme tomuto útoku predišli, je nutné používať na vkladanie informácie do údajových zhlukov iba také videá, ktoré vznikli kódovaním so statickou hodnotou parametra bitrate.

Na záver nášho výskumu sme sa zaoberali otázkou možnosti nenápadného transportu video súborov obsahujúcich ukrytú informáciu. Zamerali sme sa na verejné (multimediálne) služby internetových úložísk. Skúmali sme charakteristiky dostupných poskytovateľov tejto služby. Cieľom bolo zozbierať dostupné informácie o jednotlivých poskytovateľoch, a na základe týchto informácií analyzovať možnosť prenosu veľkých objemov dát bez vzbudenia podozrenia.

Ak sa medzi poskytovateľmi nachádzajú nejaké úložiská s podporou vlastnosti statického bitrate, môžeme ich využiť na prenos súborov, ktoré uchovávajú vloženú informáciu pomocou nami navrhnutého steganografického systému. Výsledky sú zhrnuté na obrázku č. 7.10.

Obr. 7.9: Všetky triedy binárnych postupností skúmanej množiny video súborov.

7.5 Zhodnotenie návrhu

Efektivita (meraná kapacitou) navrhnutého steganografického systému závisí od obsahu videa. Ak sa použije na prenos informácie video so statickým obrazom (napríklad s jedným objektom v popredí pri zachovaní nemenného pozadia), kapacita nosiča je veľmi malá. V našich výskumoch sme sledovali kapacitu pre 1000 videí rovnakého žánru (ženská kozmetika), ktorý bol práve typom skôr statického videa. Všetky videá mali dĺžku aspoň 14 minút. Priemerná kapacita na túto dĺžku videa činí možnosť ukrytia cca. 400 bitov (teda 50 bajtov) tajnej informácie.

Je to spôsobené tým, že v statickom videu sa nachádza oveľa menej využiteľných GOP než v dynamickom. Aj keď štandardy rodiny `MPEG` nijako priamo neobmedzujú veľkosť GOP, predsa je dĺžka GOP obmedzená. Štandard hovorí o tom, ako často sa musí vyskytnúť kľúčová snímka. Tá je kódovaná vnútro–snímkovou technikou, jedná sa o snímku typu I. A keďže snímky tohto typu definujú začiatok GOP, toto obmedzenie nepriamo určuje maximálnu dĺžku GOP, ktorá v prípade nami sledovaných videí neprekračuje hodnotu 60 (vrátane začiatočnej I snímky).

služba	FLV kontainer				MP4 kontainer			
	Formát	FPS	Bitrate (Kbps)	Rozlíšenie	Formát	FPS	Bitrate (Kbps)	Rozlíšenie
Blip.tv	VP6	24	400	1280x720				
Clipshack	H.263	30,3	458	320x240				
Dailymotion	H.263	29		320x180	AVC	Variabilné	Variabilné	512x288
Dotsub.com					AVC	Variabilné	Variabilné	420x316
Flickr					AVC	29,97	Variabilné	640x480
Glumbert	VP6	25	131	320x240				
Metacafe	AVC	30	664	640x360				
Openfilm					AVC	25	Variabilné	920x518
Sevenload	H.263	25	486	300x240				
Viddix	VP6	25	749	432x324				
Video.aol.com	H.263	25	464	468x352				
Vimeo					AVC	25	Variabilné	1280x720
Yahoo Video	VP6	30	636	400x222				
YouTube	AVC	24	1036	854x480	AVC	Variabilné	Variabilné	1920x1080

Obr. 7.10: Zoznam dostupných multimediálnych internetových úložísk [4].

Okrem tohto obmedzenia jestvuje aj obmedzenie na výslednú kvalitu obrazu. I snímka sa zavádza do video toku v prípade prekročenia miery povolenej tolerancie chyby dekódovania. Ak je teda video statického charakteru, I snímky sa budú do obrazu zavádzať iba kvôli synchronizačnému obmedzeniu, nie kvôli zvyšovaniu kvality výsledného obrazu. Naopak, v prípade videa, v ktorom sa dynamicky mení celá scéna, je nutné častejšie než raz za 60 snímok vkladať referenčnú snímku.

Teda výsledná kapacita média je silne závislá od charakteru videa a tiež od toho, či je použitý statický alebo dynamický bitrate. Ak je bitrate dynamický, kapacita média drasticky klesá, pretože na vloženie správy nemožno použiť štruktúru fyzického uloženia údajov bez vzbudenia podozrenia.

Na samotný prenos modifikovaného nosiča je možné použiť dostupné služby internetových poskytovateľov priestoru na dočasné uchovávanie multimediálneho obsahu. Medzi nimi sa nachádza viacero takých poskytovateľov, ktorí umožňujú využívať statický bitrate.

Aj keď sa zdá, že kapacita nosiča je vzhľadom na jeho veľkosť neprimerane malá, takýto steganografický systém môže slúžiť ako:

1. riadiaci systém, nakoľko môže niesť konfiguračné informácie o steganografickom systéme použitom na prenos samotných údajov v jednotlivých snímkach,
2. pomocný systém, keď môže uchovávať nejaké informácie nezávislé na hlavnom steganografickom systéme, a nakoniec aj ako
3. kontrolný systém, pretože môže uchovávať informácie umožňujúce kontrolu integrity údajov prenášaných hlavným steganografickým systémom.

Kapitola 8

Optimalizácia steganografických systémov

V súčasnosti je steganografia (ukrývanie existencie informácií) oblasťou, ktorá sa prudko rozvíja. S návrhom steganografických systémov súvisí veľká snaha o ich kompromitáciu. Hlavným nástrojom na určenie, či médium použité v obrazovom steganografickom systéme obsahuje skrytú správu, je sledovanie odchýlok od štatistických charakteristík podobných médií, ktoré správu neobsahujú.

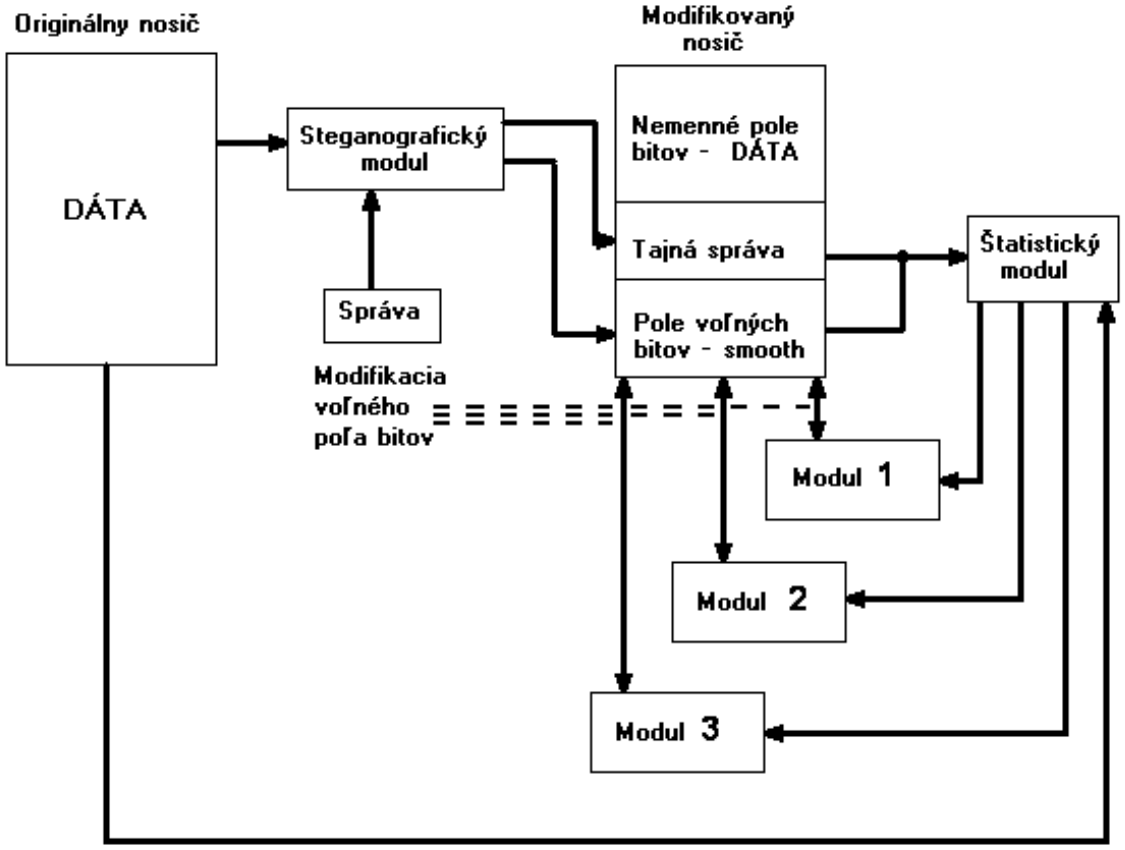
Cieľom tejto časti práce je navrhnúť niekoľko modulov steganografického systému, ktoré by sťažili útočníkovi odlíšenie média s ukrytou správou od čistého média:

- prvý modul má vyrovnávať sledované štatistiky modifikovaného média vzhľadom na čisté médium pomocou genetických algoritmov,
- druhý modul pomocou neurónových sietí a
- tretí pomocou horolezeckých (angl. *hill–climbing*) algoritmov.

Inšpiratívnou prácou pre experimenty v tejto oblasti bola [32] a predovšetkým [23]. Na práci spolupracovali [66] (časť horolezeckého algoritmu) a [8] (štatistický modul, genetické algoritmy a neurónové siete). Schéma navrhnutého systému pre vyrovnávanie štatistických vlastností nosiča po vložení správy sa nachádza na obrázku č. 8.1. Moduly označené číslami (1, 2 a 3) symbolizujú možnosť zapojenia častí, ktoré dokážu modifikovať nosič obsahujúci správu tak, aby sa isté (sledované) štatistické charakteristiky čo najmenej odlišovali od pôvodného nosiča. Pri našom výskume sme implementovali tri moduly, ktoré postupne v práci popíšeme: modul genetických algoritmov, neurónové siete a modul implementujúci horolezecký (hill–climbing) algoritmus.

8.1 Modul steganografického média

V našej práci sme sa zamerali na porovnanie efektivity viacerých techník umelej inteligencie a evolučných výpočtov. Efektivitu sme merali vzhľadom na lepšie výsledky pri úprave média obsahujúceho správu tak, aby prítomnosť tejto správy bola vzhľadom na nami definované testy nedetegovateľná.

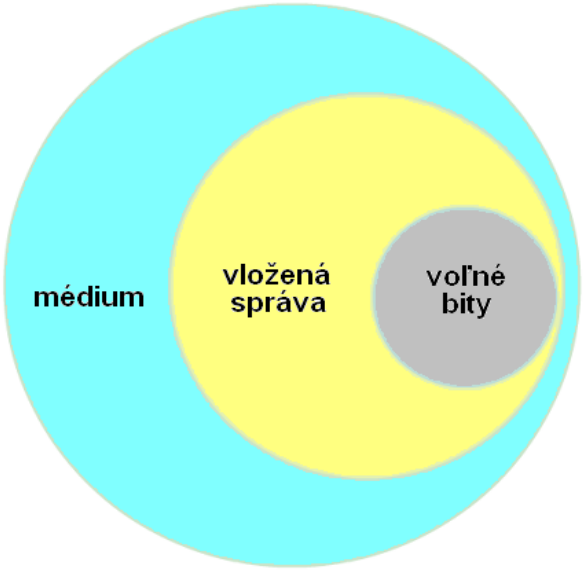


Obr. 8.1: Schéma systému pre vyrovnávanie štatistík [66].

Každý steganografický systém založený na modifikácii nosiča narúša jeho pôvodnú štruktúru. Toto narušenie je možné modelovať mnohými spôsobmi; jedným z najúčinnejších je pomocou zašumeného prenosového kanála. Cieľom steganografických útokov štatistickými metódami je tento šum nie odfiltrovať, ale odhaliť a zvýrazniť. Ak je možné nájsť odchýlku od vlastností média, ktoré steganografickú správu neobsahuje, je možné navrhnúť s cieľom kompromitovať takýto steganografický systém. Preto je cieľom steganografických systémov založených na modifikácii nosiča jeho minimálna zmena.

Ak sa aj informácia do média vkladá, zväčša sa nevyužíva celá kapacita média na vloženie správy (práve kvôli znižovaniu možnosti odhalenia vloženia správy). Redundantné informácie, ktoré bolo možné využiť na vloženie správy takto ostávajú nevyužité (hoci nezmenené).

Cieľom našej štúdie bolo experimentálne zistiť, či je možné využiť nemodifikované (ale pre steganografické účely využiteľné) informácie na kamufláž pôvodného šumu nosiča. Nakoľko sme chceli porovnávať efektivitu algoritmov samotných, nezamerali sme sa na žiaden konkrétny steganografický nosič. Nedefinovali sme žiadnu informačno–logickú štruktúru nosiča (napr. presnou špecifikáciou súborového formátu), aby sme nezaviedli možné korelácie medzi použitým algoritmom a informačnou štruktúrou, s ktorou algoritmus pracuje.



Obr. 8.2: Schéma štruktúry modelovaného steganografického nosiča [8].

Nami modelovaný steganografický nosič pozostáva z troch vzájomne disjunktných častí (viď obrázok č. 8.2):

médium: jedná sa o nemodifikovateľné oblasti; tieto reprezentujú logickú štruktúru akéhokoľvek nosiča (súborového formátu) spolu s informáciami, ktoré nie je možné modifikovať bez vysokého rizika odhalenia existencie vloženej správy;

vložená správa: táto oblasť modeluje vloženie tajnej informácie;

voľné bity: ide o oblasti využiteľné pre vloženie správy; modelujú tie časti nosiča, ktoré síce je možné využiť na vloženie správy, ale steganografický systém ich (z rôznych dôvodov, napríklad prebytočnej kapacity) nepotreboval využiť na uloženie vkladanej informácie.

Pri tvorbe takto predstaveného steganografického nosiča sme postupovali nasledovne:

Krok 1: inicializácia všetkých bitov nosiča pomocou generátora (pseudo)náhodnej binárnej postupnosti s rovnomerným rozdelením,

Krok 2: uloženie takto vytvoreného média (považuje sa za originál, ktorý v sebe neobsahuje ukrytú správu),

Krok 3: náhodný výber množiny bitov, ktoré tvoria oblasti s vloženou správou,

Krok 4: modifikácia tejto množiny bitov reprezentujúcej vloženú správu a

Krok 5: náhodný výber množiny bitov, ktoré je možné využiť pre ďalšie vkladanie správ.

Pri tvorbe steganografického nosiča je nutné zachovať pomery veľkostí jednotlivých množín bitov tak, ako je naznačené na obrázku č.8.2. Teda z pôvodného média sa vyčlení časť bitov, ktorú je možné využiť na vloženie správy. Z tejto množiny sa vyčlení časť pre samotnú správu, a časť ostane nevyužitá. Nevyužitá časť tvorí pole „voľných“ bitov, ktoré môžu jednotlivé testované algoritmy modifikovať. Jednotlivé oblasti sú iba kvôli názornosti uzavreté v súvislých blokoch.

Cieľom jednotlivých modulov, ktoré implementujú genetické, hill–climbing algoritmy a neurónové siete je hľadať také naplnenie množiny „voľných“ bitov, aby sa sledované štatistické vlastnosti pôvodného (teda média bez vloženej správy) a modifikovaného média čo najmenej odlišovali.

Sme si vedomí toho, že náš model takto definovaného nosiča nezodpovedá žiadnemu reálnemu prenosovému kanálu, pretože každý[1] komunikačný kanál prenáša informácie v štruktúrovanej podobe. Dôvody, ktoré nás viedli k tomuto modelu, sme objasnili vyššie. Modelovanie vloženej informácie pomocou rovnomerného binárneho rozdelenia je vhodné, pretože väčšina navrhnovaných steganografických systémov uvažuje s takým predspracovaním vkladanej informácie (napríklad šifrovanie), ktorá vedie na binárnu reprezentáciu s rovnomerným rozdelením núl a jednotiek.

8.2 Štatistický modul

Štatistický modul bol navrhnutý s cieľom porovnávať mieru zhody medzi dvoma rovnako veľkými blokmi údajov. Blokom dát chápeme pole bitov, ktoré je obsluhované pomocou steganografického modulu. Úlohou bolo navrhnúť tento modul tak, aby bol nezávislý od ostatných modulov, a aby bol jednoducho rozšíriteľný o ďalšie testy zhody. V prototype sme implementovali triviálne bitové porovnávanie a jeho vypešenie, ktoré je invariantné voči posunutiu po osi y.

Triviálny test zhody

Realizácia najjednoduchšieho porovnávania polí je založená na výpočte sumy rozdielov príslušných prvkov dvoch polí. Toto porovnávanie je možné robiť nielen po bitoch, ale po ľubovoľných n-ticiach bitov. Výslednú sumu normalizujeme, a jej doplnok k jednotke považujeme za mieru zhody:

$$e = 1 - \frac{1}{2^n N} \sum_{i=1}^{N} |a_i - b_i|,$$

kde e je miera zhody, N počet prvkov polí, n veľkosť bitovej reprezentácie jedného prvku poľa, a a b sú porovnávané polia.

[1] tým myslíme každý nepodozrivý vzhľadom na prenos informácie

Invariant triviálneho testu zhody

Cieľom nášho výskumu nebolo navrhovať štatistické testy zhody médií. Tie sa účinne môžu robiť pre nosiče, kde je známa fyzická štruktúra a jej väzba na informačný tok. Napriek tomu sme navrhli rozšírenie triviálneho testu zhody, aby sme mohli experimentálne overiť nezávislosť testovaných algoritmov od použitého štatistického modelu.

Základnou myšlienkou návrhu je použitie lineárnej regresie na hľadanie miery zhody. Lineárna regresia spočíva vo vyjadrení závislosti prvej premennej od druhej na bodovom grafe [68]. Pre i–ty bit vynesieme na graf bod so súradnicami (a_i, b_i), kde a a b sú porovnávané polia. Pri podobných poliach dostaneme body ležiace na diagonále. Všeobecne platí, že body budú ležat v blízkosti myslenej priamky, ktorá je určená rovnicou $a = \alpha + \beta * b$, kde α, β sú konštanty.

Problém nájdenia priamky je problémom nájdenia konštánt α a β. Na ich vyjadrenie použijeme metódu najmenších štvorcov:

$$\sum_{i=1}^{N} a_i = N * \alpha + \beta * \sum_{i=1}^{N} b_i$$

$$\sum_{i=1}^{N} a_i * b_i = \alpha * \sum_{i=1}^{N} b_i + \beta * \sum_{i=1}^{N} b_i^2,$$

kde α, β sú nami hľadané konštanty a a, b porovnávané polia.

Po získaní lineárnej charakteristiky priamky môžeme vyjadriť samotnú rozdielnosť. Na to použijeme výpočet odchýlky prvkov jednotlivých polí od získanej regresnej priamky:

$$e_i = \alpha + \beta * b_i - a_i$$

Priemernú odchýlku získame pomocou

$$\overline{e} = \frac{\sum_{i=1}^{N} |e_i|}{N}$$

Nakoľko priemerná ochýlka môže nadobúdať hodnoty z rozsahu prvkov polí, normalizovaním získame hodnotu, ktorej doplnok k jednotke považujeme za odhad miery zhody:

$$e = 1 - \frac{1}{2^n N} \sum_{i=1}^{N} |e_i|$$

8.3 Modul využívajúci genetické algoritmy

Evolučné algoritmy, a teda aj genetické, sa objavili začiatkom 60. rokov 20. storočia. Aj keď vznik genetických algoritmov sa datuje už od 60. rokov 20. storočia, ich rozšírenie a používanie vo väčšej miere sa začalo až v posledných desiatich rokoch dvadsiateho storočia. Evolučné algoritmy, čo sú stochastické optimalizačné metódy, sa dnes už bežne používajú vedcami ako aj praktickými špecialistami na riešenie celej rady problémov v najrôznejších odboroch [40].

V literatúre možno nájsť rôzne definície genetického algoritmu, ktoré sa líšia väčšinou v spôsobe vytvárania novej populácie. Geneticky algoritmus je univerzálnym stochastickým prehľadávacím prístupom, ktorý je v ohraničenom priestore prípustných riešení daného problému schopný priblížiť sa ku globálnemu optimu [72]. Všeobecná schéma algoritmu je nasledovná [40]:

Krok 1: Vynuluj hodnotu počítadla generácií $t = 0$.

Krok 2: Náhodne vygeneruj počiatocnú populáciu. Populácia predstavuje súbor reťazcov (v kontexte genetických algoritmov sa týmto reťazcom hovorí *chromozómy*, ktoré reprezentujú možné riešenia prehľadávaného priestoru. Každý chromozóm je nositeľom (reprezentantom) istých merateľných charakteristík (tzv. *génov*), ktoré podmieňujú efektivitu riešenia dosiahnuteľného daným chromozómom.

Krok 3: Vypočítaj ohodnotenie (tzv. *fitness*) každého chromozómu v počiatočnej populácii $P(0)$.

Krok 4: Vyber dvojice chromozómov z populácie $P(t)$ a vytvor ich potomkov $P'(t)$.

Krok 5: Ohodnoť novo vytvorené chromozómy $P'(t)$.

Krok 6: Vytvor novú populáciu $P(t+1)$ z pôvodnej populácie $P(t)$ a množiny potomkov $P'(t)$.

Krok 7: Zväčši hodnotu počítadla generácií o jednotku, t.j. $t := t+1$.

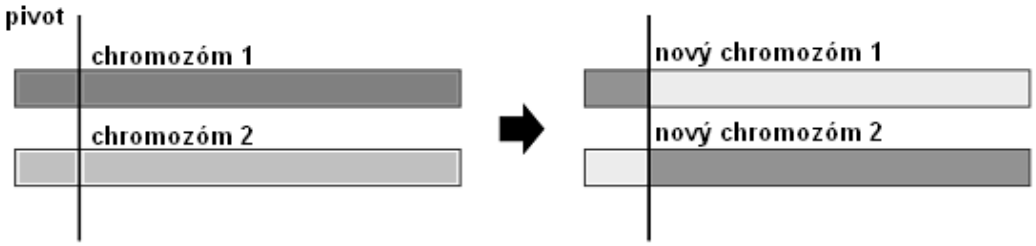
Krok 8: Pokiaľ je t rovné maximálnemu počtu generácií, alebo je splnené iné ukončovacie kritérium, vráť ako výsledok populáciu $P(t)$, inak pokračuj krokom číslo 4.

Na modifikovanie jednotlivých jedincov populácie sa používajú tzv. *genetické operácie* kríženia a mutácie.

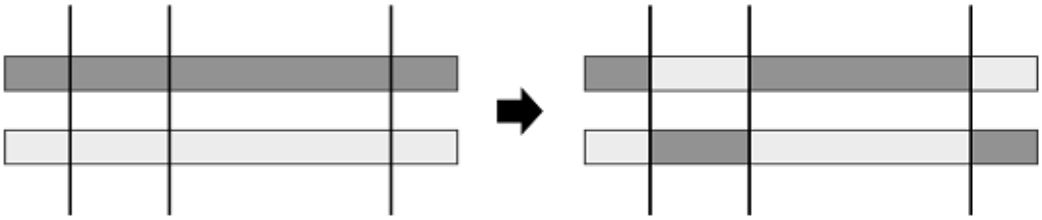
Kríženie

Medzi najznámejšie spôsoby kríženia patria nasledovné spôsoby: jednobodobé, viacbodové, maskované a diskrétne kríženie.

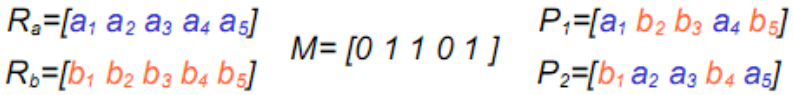
Jednotlivé chromozómy sa skladajú z usporiadaného reťazca génov. Pri jednobodovom krížení sa zvolí nejaký gén, tzv. *pivot*; ten má svoje pevne dané poradie. Nový jedinec vznikne spojením dvoch genetických reťazcov (z dvoch predkov) takým spôsobom, že z prvého predka sa zoberie genetická informácia od začiatku reťazca po pivot, a z druhého od pivotu po koniec.



Obr. 8.3: Schéma jednobodového genetického kríženia [72].



Obr. 8.4: Ukážka trojbodového genetického kríženia [72].



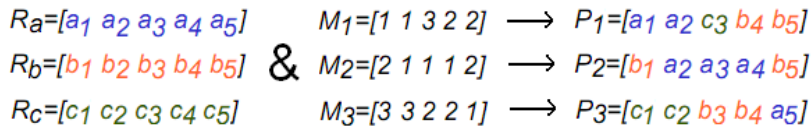
Obr. 8.5: Ukážka maskovaného kríženia [72].

Z dvoch predkov je možné pri jednom pivote vytvoriť maximálne dvoch nových jedincov. Túto situáciu názorne zobrazuje obrázok 8.3.

Viacbodobé kríženie využíva viacero pivotov naraz. Ak nám n pivotov rozdeľuje chromozóm na $m = n+1$ častí, potom pomocou dvoch chromozómov vieme vytvoriť viacbodovým krížením maximálne $2^m - 2$ nových jedincov. Ukážku vzniku dvoch nových chromozómov pri trojbodovom krížení vidno na obrázku 8.4.

Maskované kríženie je spôsobom kríženia, pri ktorom sa na výmenu jednotlivých génov použije maska, ktorá určuje, ktorý gén v chromozóme sa zamení a ktorý ostáva nezmenený. V príklade na obrázku 8.5 reprezentujú reťazce R_a a R_b dva vzájomne sa krížiace chromozómy. Reťazce P_1 a P_2 sú výsledkom kríženia a M je maska použitá na kríženie. 0 v maske určuje miesto, kde sa gény chromozómu nemenia a 1 označuje miesto, kde sa gény chromozómov navzájom vymenia.

Pri diskrétnom krížení sa jedná o maskované kríženie s r rodičmi, $r > 2$. Diskrétne kríženie popisuje schéma, v ktorej každému chromozómu (rodičovi) zodpovedá špecifický symbol v maske (napríklad identifikačné číslo chromozómu v populácii). Maska obsahuje pri každom géne špecifikáciu, z ktorého rodiča sa daný gén skopíruje. Príklad vidno na obrázku 8.6.



Obr. 8.6: Ukážka diskrétneho kríženia [72].

Mutácia

Druhou operáciou využívanou pri tvorbe nových chromozómov v genetických algoritmoch je mutácia. Mutácia môže nastať buď v rámci chromozómu alebo v rámci celej populácie. Na mutovanie chromozómov sa používajú nasledovné techniky: obyčajná, aditívna a multiplikatívna mutácia.

Nech je chromozóm nasledovne reprezentovaný reťazcom génov: $r = [r_1, r_2, \ldots, r_k, \ldots, r_n]$. Zmenu jedného génu označujeme symbolom ρ. Teda zmenu k-teho génu chromozómu r zapíšeme: $r = [r_1, r_2, \ldots, r_k, \ldots, r_n] \rightarrow r' = [r_1, r_2, \ldots, \rho, \ldots, r_n]$

Pri obyčajnej mutácii sa na ľubovolnom mieste v reťazci nahradí gén za ľubovolnú hodnotu z daného rozsahu. Teda platí, že $\rho \in < r_{k_{min}}; r_{k_{max}} >$. Aditívna mutácia mierne modifikuje hodnotu génu tak, že k danej hodnote pripočíta náhodné číslo ϵ z vopred zvoleného intervalu. Výsledná hodnota nesmie neprekročiť povolený rozsah hodnôt mutovaného génu: $\rho = r_k + \epsilon$. Multiplikatívna mutácia zavádza nelineárny prvok; hodnota mutovaného génu sa vynásobí náhodne zvoleným číslom δ tak, aby výsledná hodnota spadala do povoleného intervalu hodnôt mutovaného génu. Teda $\rho = r_k * \delta$.

Výber nových jedincov populácie

Pri výbere jedincov je dôležité zachovať princíp evolúcie. To znamená, že „silnejší“ jedinec má väčšiu šancu prežit — dostať sa do novej populácie. Nesmieme však zabúdat aj na to, že musí byť zachovaná dostatočná rozmanitosť jedincov. Ak bude totiž preferované iba kritérium výberu najsilnejších jedincov, môže sa stať, že jedinci s relatívne vyšším hodnotením prevládnu v celej populácii. To bude viesť k riešeniu, ale ak problém obsahuje viacero lokálnych maxím, a systém bude konvergovať v takomto stave k niektorému z nich, nebude možné nájsť lepšie riešenie. Tak bude populácia konvergovat k suboptimálnemu riešeniu [40].

V genetických algoritmoch sa používajú viaceré metódy a techniky na výber chromozómov z populácie. V tejto časti popíšeme iba základné metódy. Nebudeme sa zaoberať upravenými metódami, nakoľko sa zakladajú na jednoduchých základných modeloch. Medzi základné mechanizmy výberu patria:

- výber na základe úspešnosti,

- náhodný výber,
- ruletový výber a
- turnajový výber.

Výber na základe úspešnosti vyžíva normalizované hodnoty ohodnotenia jednotlivých jedincov populácie. Ohodnotenie sa chápe ako pravdepodobnosť, s ktorou sa daný jedinec dostane do nasledujúcej populácie.

Náhodný výber je realizovaný (pseudo)náhodnou funkciou, ktorá vyberá jedincov novej populácie. Tým sa zabezpečuje vysoká variabilita jedincov v rámci všetkých populácií.

Ruletový výber je jedným z najčastejšie používaným. Z jednotlivých ohodnotení všetkých jedincov populácie sa vypočíta suma. Všetky ohodnotenia jedincov sa normujú vzhľadom na túto sumu. Ruletový výber spočíva v pripodobnení výberu jedincov do novej populácie rulete. Suma ohodnotení tvorí kruh s jednotkovým obsahom. Jednotlivé normované ohodnotenia sú reprezentované výsekmi kruhu s príslušnými obsahmi. Výber sa realizuje náhodným výberom na takto vytvorenom ruletovom kolese. Jedince s väčším ohodnotením majú vyššiu šancu dostať sa do ďalšej populácie než jedince s nízkym ohodnotením. Ak je ohodnotenie nejakého tvorí viac než 90% obsahu ruletového kruhu, je vhodné použiť tzv. váhovaný ruletový výber. Ten spočíva v pridelení váh jednotlivým jedincom, aby sa dosiahlo približne rovnomerné rozdelenie výberu.

Pri turnajovom výbere sa náhodným spôsobom vyberá skupina jedincov z pôvodnej populácie. Následne sa z tejto podmnožiny selektuje isté percento najlepších jedincov. Nakoľko kríženie prebieha až po výbere rodičovských jedincov, a na realizáciu výberu je potrebná vždy aspoň dvojica rodičov, náhodný výber podskupiny sa robí dva krát. Po realizovaní selekcie sa môže začať proces kríženia a mutácie, čím sa vytvorí nová generácia jedincov. Výhoda tohto výberu je v tom, že nepreferuje ani najsilnejších, ani najslabších jedincov. Tým vytvára konštantný tlak selekcie v každej populácii, a preto môže súčasne nachádzať viacero riešení (pri problémoch, ktoré nemajú iba jedno riešenie).

Nastavenie parametrov algoritmu

Správanie sa genetických algoritmov nie je závislé iba na konkrétnom algoritme, použitých genetických operáciách a výberoch. Pre dosiahnutie optimálnych výsledkov je nutné vhodne nastaviť niektoré parametre genetického algoritmu, aby vyhovovali riešenému problému. Jedná sa predovšetkým o tieto nastavenia: pravdepodobnosť kríženia, pravdepodobnosť mutácie, počet generácií, ukončovacia podmienka a veľkosť populácie.

Ukazuje sa, že nie je jednoduché nájsť explicitné vyjadrenia týchto nastavení, nakoľko vzhľadom na správanie sledovaného systému nie sú nezávislé. V literatúre je možné nájsť niekoľko odporúčaných postupov pri odhade hodnôt týchto parametrov [85]:

- použitie odporúčaných hodnôt,
- teoretické odvodenie hodnôt,
- empirické nastavenie hodnôt a
- samoadaptívnosť.

Hoci je vhodné pre špecifický problém hľadať optimálne nastavenia parametrov, takmer pre všetky úlohy riešené pomocou genetických algoritmov sa dajú využiť odporúčané hodnoty uvedené v tabuľke 8.1. V prvom stĺpci sa nachádza typ parametra, v druhom príslušné hodnoty podľa Dejonga [19] a v poslednom podľa Grefenstette [33].

Pri istých problémoch, ktoré sa riešia pomocou genetických algoritmov, je možné teoreticky odvodiť ohraničenia hodnôt niektorých parametrov, napríklad veľkosti populácie. Úvah a prístupov, ktoré vedú pri tom istom probléme ku konkrétnym hodnotám, je viacero. Navyše, vo svojich odporúčaniach ohľadom hodnôt parametrov sa často nezhodujú, a preto nie je celkom zrejmé, ktorý z teoretických výsledkov aplikovať.

Najčastejšou metódou pri optimalizácii parametrov gnetických algoritmov je ich empirické zisťovanie. Ide o sadu experimentov, v rámci ktorej sa postupne menia hodnoty sledovaných parametrov, a podľa stanovenej metodiky sa vyhodnocuje efektivita systému pri danom nastavení. Táto metóda je časovo veľmi náročná, a nemusí viesť k optimálnemu nastaveniu. Dôvodom je spomínaná skrytá korelácia jednotlivých parametrov a v istých prípadoch silná citlivosť systému na zmenu niektorých nastavení.

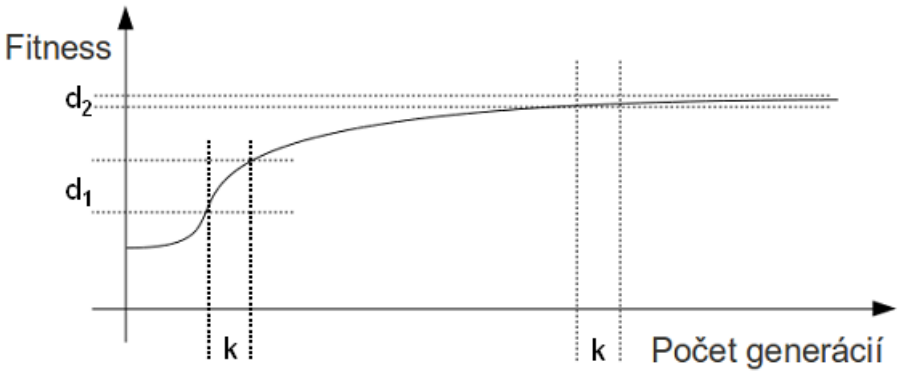
Tabuľka 8.1: Prehľad odporúčaných parametrov genetických algoritmov.

parameter	Dejong [19]	Grefenstette [33]
veľkosť populácie	50	30
počet generácií	1000	—
typ kríženia	dvojbodové	dvojbodové
pravdepodobnosť kríženia	0.6	0.9
typ mutácie	zámena bitov	zámena bitov
pravdepodobnosť mutácie	0.001	0.01

V nami navrhnutom module používame nastavenia prevzaté od Grefenstette [33]. Konkrétne hodnoty dôležitých parametrov možno vidieť v tabuľke 8.2. Na ukončenie algoritmu neslúži fixne daný počet generácií, ale sleduje sa zlepšovanie výsledku. Ak už nie je možné dosiahnuť pokrok, algoritmus končí. Na sledovanie tohto zlepšovania slúžia ohodnotenia posledných 100 generácií.

Tabuľka 8.2: Hodnoty parametrov genetických algoritmov použitých v našom návrhu.

parameter	naše nastavenia
veľkosť populácie	30
počet generácií	—
typ kríženia	dvojbodové
pravdepodobnosť kríženia	0.9
typ mutácie	zámena bitov
pravdepodobnosť mutácie	0.01
percento prekrytia populácií	0.4
počet sledovaných generácií	100



Obr. 8.7: Ukážka znižovania miery rastu vo výpočte [8].

Návrh modulu

Na vyrovnanie sledovanej štatistiky steganografického média sme navrhli dva genetické algoritmy. Obidva vychádzajú z rovnakého modelu a líšia sa iba v použitom výbere (selekcii). Implementovaný model navyše využíva prekrývajúce sa populácie. V každej populácii umožňuje urciť množstvo jedincov, ktoré budú nahradené novými. Genetický algoritmus č. 1 používa na vytvorenie novej populácie turnajový výber, algoritmus č. 2 ruletový výber. Oba algoritmy majú rovnakú ukončovaciu podmienku, a to mieru konvergencie. Ak sa ohodnotenie nezlepšuje (v algoritme daný) istý počet populácií, výpočet končí (viď obrázok č. 8.7).

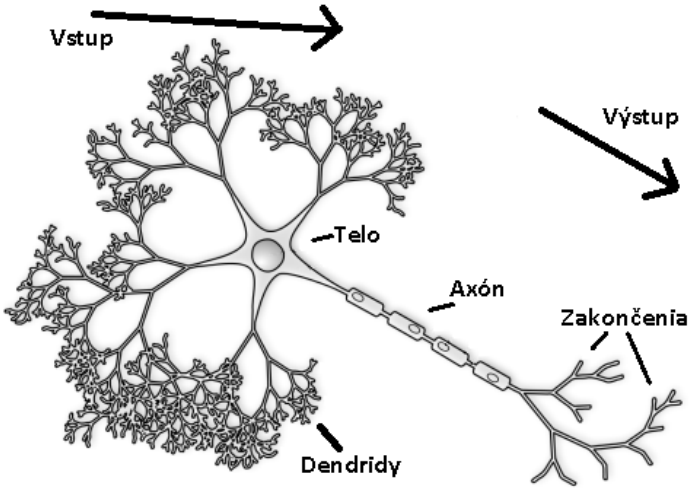
8.4 Modul využívajúci neurónové siete

Neurónové siete sú matematickým modelom činnosti mozgu. Ľudský mozog obsahuje približne 10 biliónov neurónov a asi 60 triliónov prepojení [38]. Mozog je vysoko komplexný, nelineárny, a paralelný systém spracovania informácií [60]. Základnou stavebnou jednotkou mozgu je neurón s nasledovnými schopnosťami:

1. prijímať signál z okolitého prostredia,
2. spracovať signál a
3. odoslať signál do prostredia.

Schému neurónu zobrazuje obrázok číslo 8.8. V ľavej časti schémy sa nachádza telo neurónu so vstupnými výbežkami, v pravej časti výstupný výbežok s rozvetvením do vstupov ďalších neurónov. V matematickom modeli neurónu sa používa neurón, ktorý má buď jeden, alebo viacero vstupov, a jediný výstup (rozvetvený na vstupy nasledujúcich neurónov). V organickej neurónovej sieti jestvujú navyše neuróny s viacerými axónmi.

Signál na vstupe neurónu je *prahovaný*. To znamená, že keď intenzita vstupného signálu nepresiahne istú hodnotu, vstup nie je aktivovaný a neurónom nie je spracovaný. Po aktivovaní vstupov (na základe presiahnutia prahovej úrovne intenzity signálu) sa tieto spracujú.



Obr. 8.8: Schéma neurónu [67].

Zväčša sa jedná o jednoduchú sumáciu s váhovaním jednotlivých vstupov. Po spracovaní sa výstupná hodnota vyšle na výstup neurónu, ktorý môže byť pripojený k ďalším vstupom iných neurónov.

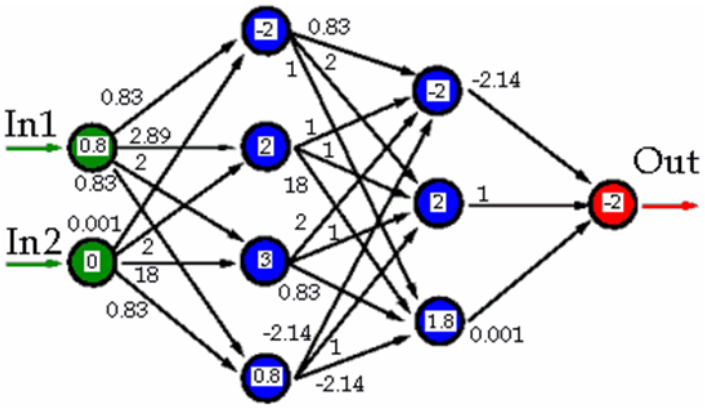
Spracovanie signálu biologickým neurónom je oproti hardvérovej realizácii pomalšie o sedem až osem rádov, a vďaka Murphyho zákonu o každoročnom zdvojnásobovaní výkonu hardvérovej architektúry počítačov sa tento pomer rýchlo zvyšuje. No napriek tejto výhode hardvéru stále nie je možné simulovať činnosť ľudského mozgu. Je to spôsobené mohutnosťou biologickej neurónovej siete — relatívne pomalá činnosť ľudského mozgu je vyvážená jeho robustnosťou (viac informácií viď napríklad v [60]).

Typy neurónových sietí

Na prvý pohľad sa neurónová sieť javí ako zhluk neurónov, ktoré sú vzájomne poprepájané. Neurónové siete sú nástrojom, ktorý počas svojej činnosti prirodzene využíva paralelné výpočty. Paralelizmus siete sa pri softvérovej (programovej) realizácii musí často iba simulovať. Paralelné spracovanie vstupného signálu bolo donedávna možné (zväčša) iba pri hardvérovej implementácii. S rozvojom technológie využívania grafických procesorov na všebecné výpočty (viď napríklad [58]) sa dá šírenie vzruchu simulovať paralelne aj softvérovo.

Viacvrstvové dopredné siete majú schopnosť aproximovať ľubovoľné spojité vstupno–výstupné mapovanie požadovaným stupňom presnosti, ak majú dostatočné množstvo skrytých neurónov (prevzaté z [60]).

Neurónová sieť je daná neurónmi a ich vzájomným prepojením. V oblasti informatiky je implementovaná buď elektronickým obvodom, alebo simuláciou pomocou programu. Formálne je možné definovať neurónovú sieť viacerými spôsobmi v závislosti od dôrazu na štruktúru, ktorá je vhodnejšia na riešenie problematiky. V literatúre (napríklad viď [49], [38], [60], [39], [48]) sa formálne neurónová sieť definuje zväčša pomocou teórie grafov alebo analogicky k činnosti ľudského mozgu.



Obr. 8.9: Ukážka doprednej neurónovej siete.

Zelenou farbou je označená vrstva vstupných vrcholov (neurónov) V_i, modrou množina skrytých neurónov V_h a červenou je označený výstupný neurón (z množiny V_o). Šípky naznačujú smer šírenia signálu.

Podľa [38] jestvujú štyri základné architektúry neurónových sietí:

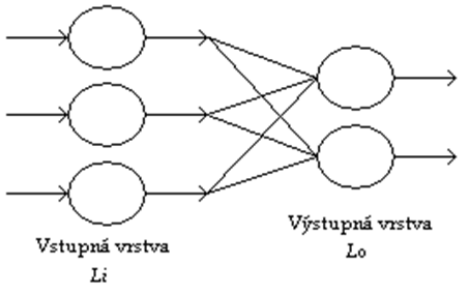
1. jednovrstvové dopredné siete,
2. viacvrstvové dopredné siete,
3. rekurentné siete a
4. mriežkové štruktúry.

Neurónové siete umožňujú upravovať, uchovávať a spracovávať vstupné informácie podobne, ako sa to deje pri biologických neurónoch, teda pomocou nastavenia ohodnotenia prepojení (hrán grafu — ako vidno aj na ukážkovom obrázku č. 8.9) medzi jednotlivými uzlami grafu (neurónmi).

Dopredná neurónová sieť pozostáva z troch množín neurónov: vstupných, skrytých a výstupných neurónov. Jednotlivé množiny tvoria takzvané vrstvy. Skrytá vrstva môže obsahovať ľubovoľný (teda aj nulový) počet podvrstiev.

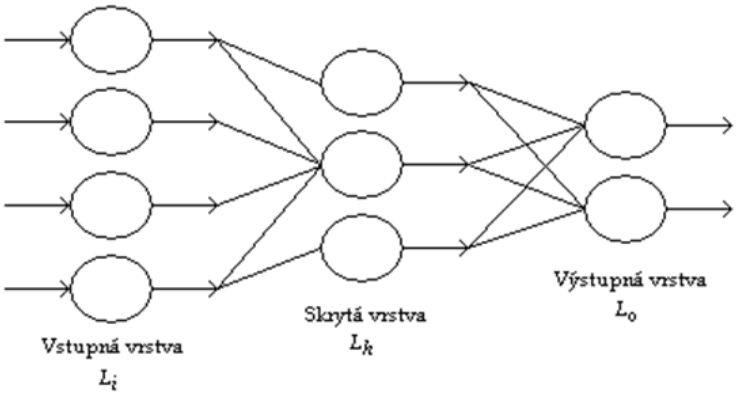
Aktivity neurónov vo vrstve L_j, $j > 1$ je možné spočítať, ak poznáme aktivity z vrstiev L_k, $k \in \{1, \ldots, j\}$. Takto postupne je možné spočítať aktivity neurónov všetkých vrstiev. Ako posledné sa počítajú aktivity neurónov výstupnej vrstvy. Výpočet prebieha od vrstvy 1 po vrstvu n — z tejto vlastnosti sa odvodzuje názov pre tento typ štruktúry: neurónové siete s dopredným šírením. Na ich reprezentáciu sa využíva acyklický orientovaný graf.

Jednovrstvová neurónová sieť používa na výpočet iba výstupnú vrstvu neurónov L_o. Vstupná vrstva slúži iba na prenos vstupných signálov k neurónom výstupnej vrstvy. Ak je výstup každého neurónu vo vrstve i spojený so vstupom každého neurónu vo vrstve $i+1$, hovoríme, že sa jedná o *úplne prepojenú doprednú neurónovú sieť*. Príklad jednovrstvovej úplnej doprednej neurónovej siete sa nachádza na obrázku číslo 8.10. Ukážka **viacvrstvovej doprednej neurónovej siete** sa nachádza na obrázku č. 8.11.



Obr. 8.10: Príklad jednovrstvovej úplnej doprednej siete.

Vstupná vrstva neurónov prenáša signál na výstupnú (výpočtovú) vrstvu siete.



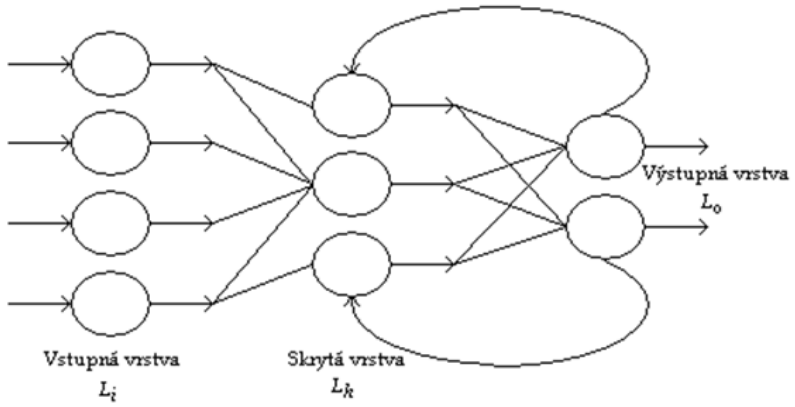
Obr. 8.11: Príklad viacvrstvovej doprednej siete.

Vstupná vrstva neurónov prenáša signál na výstupnú (výpočtovú) vrstvu siete pomocou jednej skrytej vrstvy.

Ak je na reprezentáciu neurónovej siete nutné použiť cyklický orientovaný graf, na výpočet aktivít neurónov je nutné použiť iteračný postup: z počiatočných aktivít sa vypočítajú nové aktivity neurónovej siete, ktoré sa stanú vstupnými aktivitami výpočtu v nasledujúcom iteračnom kroku algoritmu. Algoritmus končí, ak je rozdiel medzi aktivitami z predošlého a aktuálneho kroku výpočtu menší, než vopred stanovená presnosť.

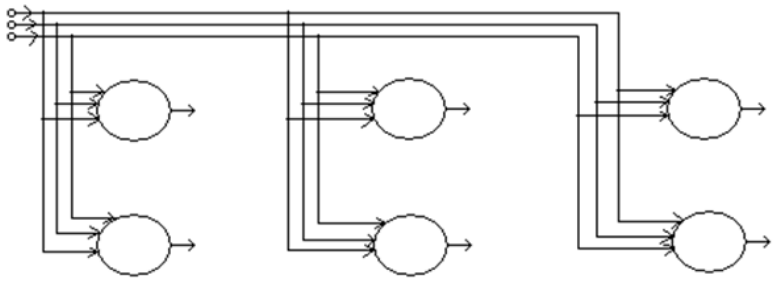
Rekurentné neurónové siete sa teda od dopredných líšia tým, že majú aspoň jednu *spätnú väzbu* [60]. Spätná väzba môže byť zavedená z ľubovoľného neurónu vo vrstve i do neurónu vo vrstve j, $j \in \{1, \ldots, i\}$. Príklad takejto siete sa nachádza na obrázku č. 8.12.

Mriežka pozostáva z aspoň jednorozmerného poľa neurónov, spolu so vstupnými signálmi, ktoré sú napojené na každý neurón v poli (porov. [60]). Dôležitou vlastnosťou tejto štruktúry je spojenie každého vstupného signálu s každým neurónom v mriežke. Jedná sa vlastne o úplnú jednovrstvovú doprednú sieť, až na usporiadanie výstupných neurónov — tie sú zoradené do mriežky. Ukážka neurónovej siete s usporiadaním výstupov do mriežky sa nachádza na obrázku č. 8.13.



Obr. 8.12: Príklad rekurentnej neurónovej siete.

Obsahuje dve spätné väzby z výstupných neurónov do skrytej vrstvy.



Obr. 8.13: Ukážka jednovrstvovej doprednej neurónovej siete.

Výstupná vrstva je usporiadaná do dvojrozmernej mriežky. Signál sa šíri v smere šípok z ľavého horného rohu obrázka.

Návrh modulu

Modul neurónových sietí je postavený na implementácii doprednej neurónovej sieti s jednou skrytou vrstvou. Implementovali sme dva typy sietí:

1. s 9 neurónmi na vstupe, 3 v skrytej vrstve a 1 na výstupe, a
2. s 5 neurónmi na vstupe, 4 v skrytej vrstve a 2 na výstupe.

Podobne, ako pri genetických algoritmoch, aj v prípade neurónových sietí je ukončovacou podmienkou nie počet opakovaní, ale miera zlepšovania výsledku. Vzhľadom na pamäťové nároky reprezentácie neurónovej siete sa sleduje posledných 30 modifikácií nastavenia váh.

8.5 Modul využívajúci horolezecký (hill–climbing) algoritmus

Technika horolezeckého algoritmu patrí medzi najjednoduchšie spomedzi všetkých v oblasti evolučných výpočtov. Je možné povedať, že ostatné techniky sa od tejto líšia iba rozšíreniami a reprezentáciou interných údajov algoritmu.

Podobne ako aj ostatné algoritmy, používa sa pri riešení problematiky, ktorá obsahuje viacero (lokálnych) extrémov. Tiež je tu skryté nebezpečenstvo, že výpočet uviazne v lokálnom extréme, a lepšie riešenie nenájde. O tom budeme hovoriť v nasledujúcej časti.

Schéma činnosti tohto algoritmu je nasledovná. Na začiatku za urobí inicializácia riešenia pomocou generátora (pseudo) náhodnej binárnej postupnosti. Hlavná funkcionalita algoritmu je vykonávaná v dvojitom cykle.

Vstupom vnútorného cyklu je inicializačný vektor riešenia. V tomto cykle sa postupne modifikuje vektor riešenia (napríklad metódou najmenších zmien) a hľadá sa taký vektor, ktorý bude ohodnotený lepšie než inicializačný. Výstupom vnútorného cyklu je buď pôvodný vektor, alebo taký, ktorý má lepšie ohodnotenie.

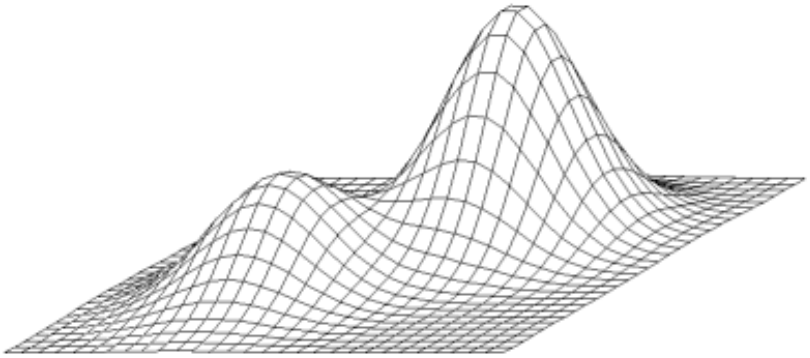
Vonkajší cyklus kontroluje postup prehľadávania. Pri splnenej ukončovacej podmienke (napríklad počet prehľadaných bodov priestoru alebo podobne ako pri predošlých technikách nízka hodnota miery konvergencie) sa proces prehľadávania ukončí. Ak podmienka nie je splnená, aktuálny vektor riešenia sa stáva novým inicializačným vektorom, a algoritmus pokračuje vykonávaním vnútorného cyklu prehľadávania okolitých bodov priestoru.

Problém lokálneho extrému

Ako bolo spomenuté v úvode, algoritmus ľahko dosahuje pri prehľadávaní priestoru lokálne extrémy. V štandardnej, t.j. najjednoduchšej verzii nie je schopný z dosiahnutého extrému robiť pokroky k lepším riešeniam. Ak by sme aplikovali tento algoritmus na funkciu, ktorej grafickú reprezentáciu znázorňuje obrázok č. 8.14 a ako prvý by bol dosiahnutý menší extrém, výpočet by skončil. Možné riešenie tohto problému uvedieme pri rozšíreniach algoritmu.

Druhým vážnym problémom môže byť zacyklenie výpočtu. Ak sa nachádzajú vedľa seba[2] minimálne dva extrémy s rovnakými hodnotami, algoritmus bude ponúkať ako konečné riešenie niektoré z nich. Pritom môže ísť znovu o uviaznutie v lokálnom extréme, len v trošku inej podobe.

[2] z pohľadu algoritmu musí ísť o susedné body prehľadávaného priestoru



Obr. 8.14: Funkcia s dvoma extrémami [66].

Rozšírenia algoritmu

V praxi sa zväčša nepoužíva základná verzia algoritmu, ale niektoré jej vylepšenia. Je to predovšetkým z časového dôvodu (prehľadanie celého priestoru je často v reálnom čase pri súčasnej výpočtovej technike nemožné) alebo z dôvodu riešenia problematiky uviaznutia v lokálnom extréme. V literatúre sa môžeme najčastejšie stretnúť s týmito typmi úprav:

jednoduchý algoritmus — neprehľadáva sa celé okolie (všetky susedné body priestoru) kandidáta na najlepšie riešenie, ale akonáhle sa nájde riešenie s lepším ohodnotením, než je aktuálne najlepšie, vnútorný cyklus algoritmu končí;

postupne stúpajúci algoritmus — základná verzia algoritmu, pri ktorej je postupne prehľadávaný celý priestor (vždy všetky susedné body aktuálne najlepšieho riešenia);

stochastický algoritmus — urýchľuje prehľadávanie tým, že zo všetkých možných susedov aktuálne najlepšieho riešenia vyberie jedného kandidáta a na základe výsledku ohodnotenia rozhodne, či sa novým najlepším riešením stane testovaný kandidát, alebo sa bude testovať ďalší náhodne zvolený susedný bod doterajšieho dosiahnutého riešenia;

algoritmus s návratom — umožňuje skúmať aj také body priestoru, ktoré dočasne (t.j. do istej v algoritme špecifikovanej hĺbky) vedú k horšiemu ohodnoteniu než aktuálne najlepšie riešenie; táto modifikácia znižuje možnosť uviaznutia v lokálnom extréme, ale za cenu vysokých pamäťových nárokov (je potrebné uchovávať dodatočné informácie umožňujúce návrat k riešeniu, z ktorého sa vyšlo);

algoritmus s náhodnou inicializáciou — patrí medzi najzložitejšie a najnáročnejšie zo všetkých spomenutých. Jedná sa o (súčasné) spustenie viacerých inštancií niektorej z vyššie spomenutých verzií algoritmu. Táto modifikácia sleduje výstupy behov jednotlivých inštancií, a z nich nakoniec vyberá riešenie s najvyšším ohodnotením. Podobne ako modifikácia algoritmu s návratom umožňuje úspešné hľadanie riešenia aj v prípade priestoru, ktorý obsahuje viacero lokálnych extrémov.

Návrh modulu

Na ukážku a porovnanie s ostatnými modulmi sme implementovali dva najjednoduchšie typy algoritmu: základnú verziu a zjednodušenú (v prípade zjednodušenej sa neprehľadávajú všetci susedia aktuálne najlepšieho riešenia, ale hľadanie končí po nájdení prvého lepšieho). Pri experimentoch nás zaujímal časový rozdiel behu jednotlivých verzií algoritmu, a taktiež efektivita meraná pomerom čas versus ohodnotenie nájdeného riešenia.

8.6 Metodika experimentov

Na účely testovania sme vytvorili špeciálny testovací modul, ktorý simuloval správanie systému, ktorého schéma sa nachádza na obrázku č. 8.1 na strane 105.

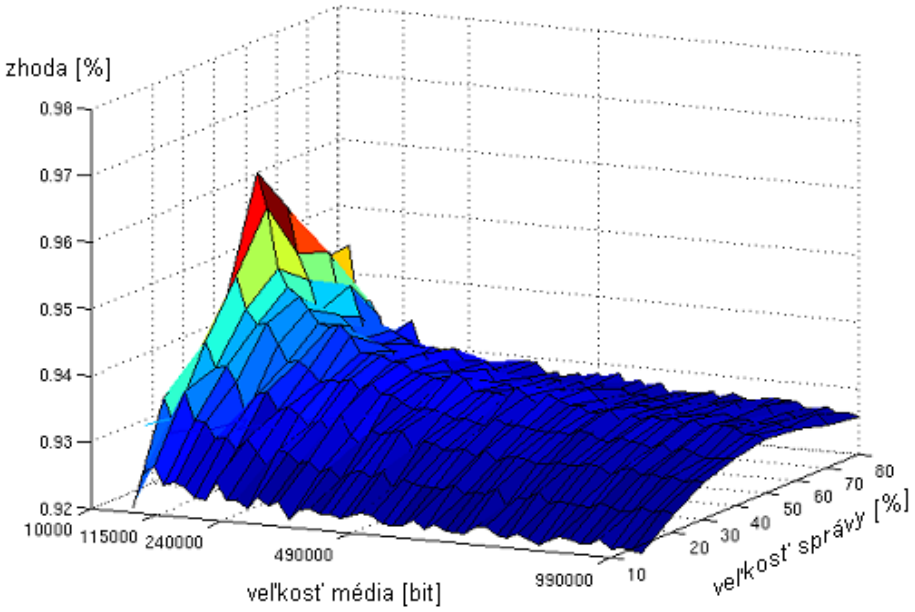
Skúmali sme závislosť ohodnotenia najlepšieho riešenia jednotlivých algoritmov od veľkosti nosiča a od veľkosti správy v nosiči. Prehľad nastavení parametrov testov sa nachádza v tabuľke č. 8.3.

Tabuľka 8.3: Prehľad nastavení parametrov testovania.

parameter	*hodnota*
počiatočná veľkosť nosiča	10 000 (bitov)
koncová veľkosť nosiča	990 000 (bitov)
krok zväčšovania nosiča	20 000 (bitov)
kapacita vyhradená pre správu	1/8 veľkosti nosiča
počiatočná veľkosť správy	10% kapacity pre správu
koncová veľkosť správy	80% kapacity pre správu
krok zväčšovania správy	10%
počet testov pre každé nastavenie	100

Výsledný súbor údajov obsahuje dvojrozmernú maticu. Každý riadok zodpovedá jednej veľkosti nosiča. Stĺpce zodpovedajú meniacej sa veľkosti správy v nosiči. Prvkami matice je ohodnotenie najlepšieho riešenia; teda takého riešenia, ktorého sledované štatistiky sa čo najviac zhodujú s nosičom, ktorý neobsahuje steganografickú informáciu.

Kapacita vyhradená pre správu činí jednu osminu celkovej veľkosti média. Túto hodnotu sme zvolili ako konštantu pre všetky testy, a vychádza z reálneho používania steganografických systémov. Pri zvyšovaní kapacity nad túto mieru sa prudko zvyšuje možnosť odhalenia vloženia správy. Nakoľko základná steganografická technika (algoritmus LSB) využíva najmenej významný bit na prenos informácie, a väčšina steganografických médií skúmaných v tejto oblasti je bajtovo orientovaná, je možné na kapacitu správy vyhradiť maximálne jeden bit z každého bajtu, čo činí práve nami definovanú kapacitu správy.



Obr. 8.15: Závislosť miery zhody od veľkosti nosiča a správy pre genetický algoritmus č. 1.

8.7 Výsledky experimentov

Výsledky testov pre genetické algoritmy sa nachádzajú na obrázkoch č.8.15 a 8.16. Na všetkých grafoch, ktoré znázorňujú závislosť `zhody` od rôznych parametrov systému, sa pod zhodou chápe percentuálna miera zhody sledovaných štatistických vlastností medzi médiom, ktoré neobsahuje steganografickú správu, a nosičom, ktorý túto správu obsahuje.

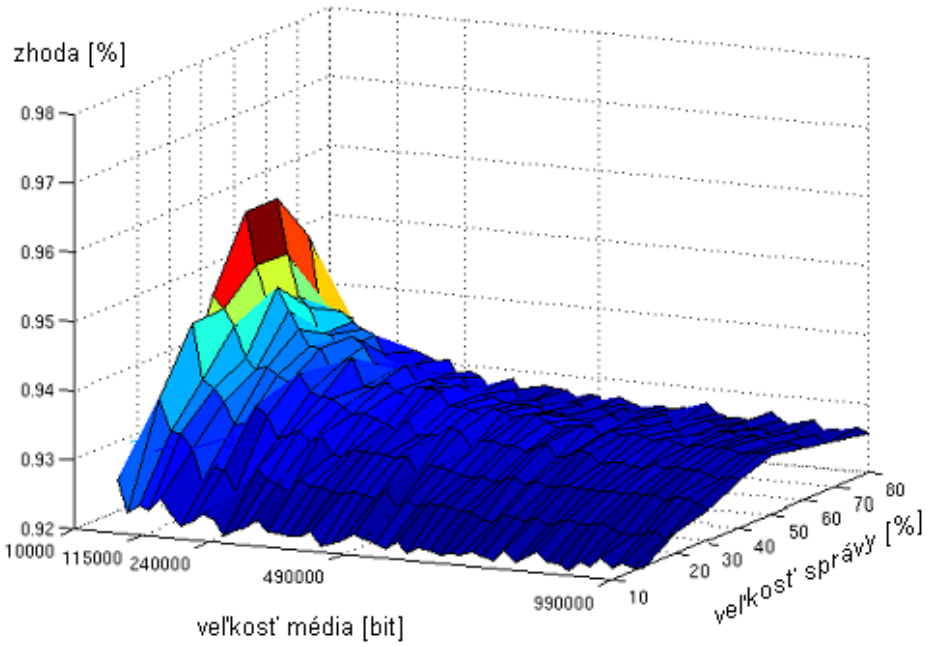
Z obrázkov vidno, že zvolené algoritmy dosahujú veľmi podobné výsledky. Algoritmy majú rovnaké parametre, líšia sa vo funkcii výberu nových jedincov do ďalšej populácie.

Najlepšie výsledky sa dosahujú pri 50% využití kapacity určenej pre správu. Rozptyl hodnôt činil 0.04%.

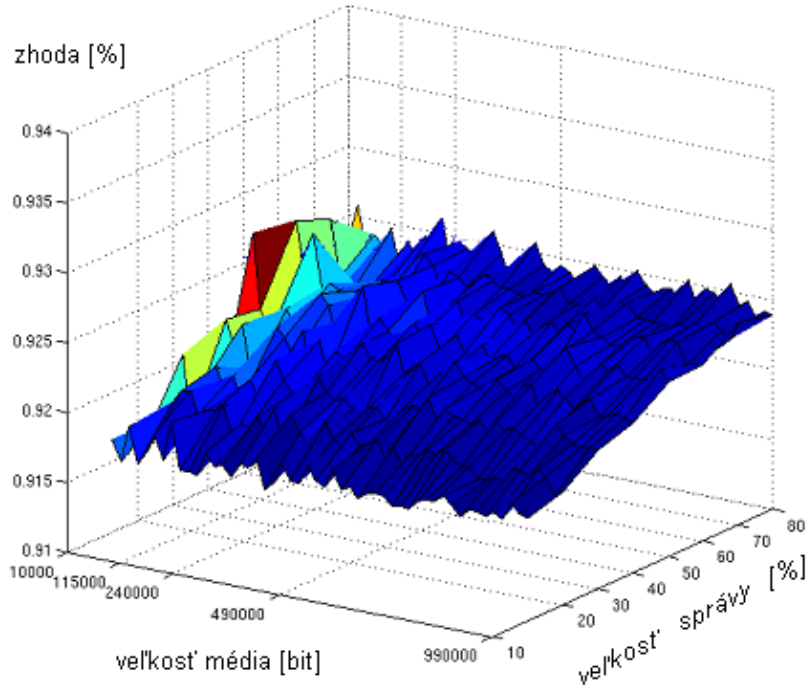
Neurónové siete, na rozdiel od genetických algoritmov, vykazujú odlišné výsledky v závislosti od štruktúry siete. Výsledky pre neurónovú sieť s 9 neurónmi na vstupe, troma v skrytej vrstve a jedným výstupným neurónom možno vidieť na obrázku č. 8.17. Značne odlišné správanie siete s piatimi vstupnými neurónmi, štyrmi v skrytej vrstve a dvoma na výstupe vidno na obrázku č. 8.18. Architektúra neurónovej siete má veľký vplyv na výsledky. V prípade neurónových sietí bol rozptyl hodnôt 0.02%.

V porovnaní s genetickými algoritmami bola efektivita neurónových sietí nižšia. Kým ohodnotenie genetických algoritmov sa pohybovalo v rozmedzí miery zhody od 0.9184 po 0.9615, neurónové siete dosahovali mieru zhody v intervale od 0.9190 po 0.9375.

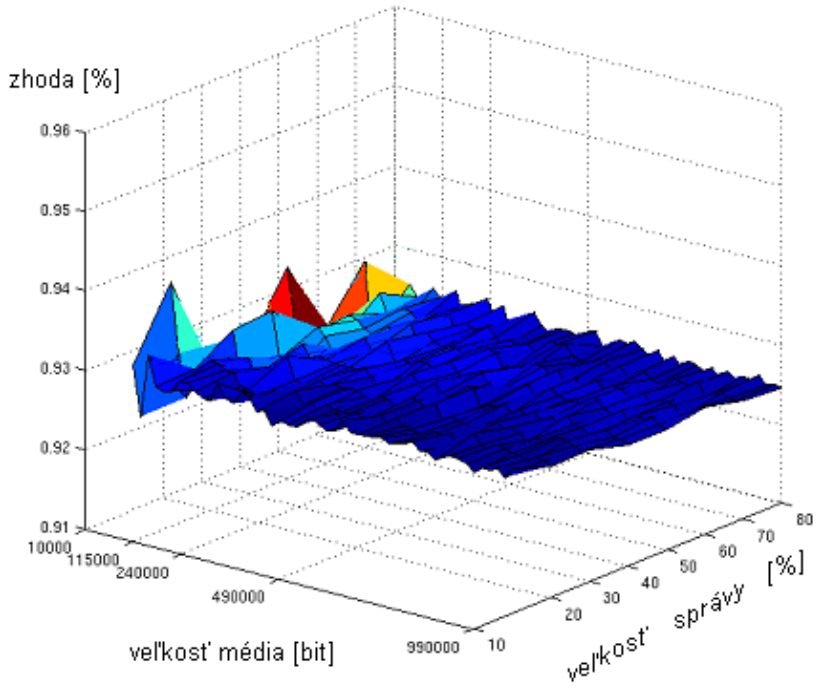
Pre porovnanie efektivity jednotlivých algoritmov sme implementovali jednoduchý testovací modul, ktorý nerobil žiadne sofistikované vyrovnávanie sledovaných štatistík. Použili sme generátor (pseudo)náhodnej postupnosti bitov, a touto postupnosťou sme v jednotlivých testoch napĺňali pole voľných bitov. Výsledky tohto náhodného vypĺňania ukazuje obrázok č. 8.19. Možno si všimnúť, že dosiahnutá hodnota miery zhody (0.75%) je rovnaká pre všetky situácie, a radikálne sa odlišuje od úspešnosti študovaných optimalizačných metód.



Obr. 8.16: Závislosť miery zhody od veľkosti nosiča a správy pre genetický algoritmus č. 2.



Obr. 8.17: Závislosť miery zhody od veľkosti nosiča a správy pre neurónovú sieť s 9 neurónmi na vstupe, 3 v skrytej vrstve a jedným na výstupe.



Obr. 8.18: Závislosť miery zhody od veľkosti nosiča a správy pre neurónovú sieť s 5 neurónmi na vstupe, 4 v skrytej vrstve a dvoma na výstupe.

Napokon sme sledovali výkonnosť horolezeckého algoritmu v jeho dvoch modifikáciách. Obrázok č. 8.20 zhŕňa výsledky pre obidve verzie a dve štatistické funkcie vzhľadom na dosiahnutú mieru zhody:

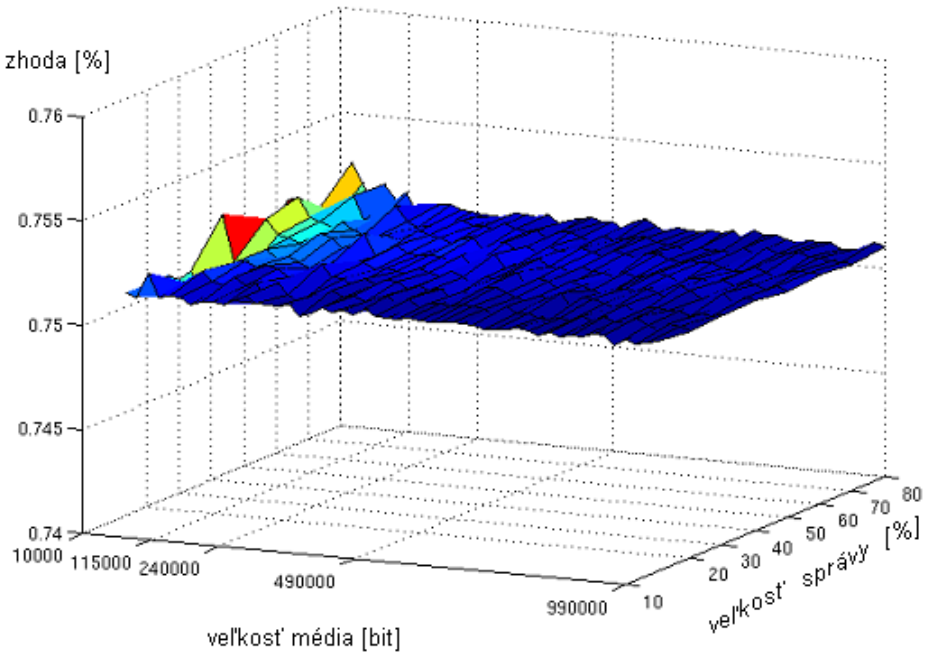
Simple_HD — zjednodušená verzia horolezeckého algoritmu (ukončuje prehľadávanie susedov po nájdení prvého lepšieho) v kombinácii so sledovaním miery zhody bitovým porovnávaním;

Steepest_HD — základná verzia algoritmu v kombinácii so sledovaním miery zhody pomocou bitového porovnávania;

Simple_SE — zjednodušená verzia horolezeckého algoritmu (ukončuje prehľadávanie susedov po nájdení prvého lepšieho) v kombinácii so sledovaním miery zhody lineárnou regresiou;

Steepest_SE — základná verzia algoritmu v kombinácii so sledovaním miery zhody pomocou lineárnej regresie.

Z uvedeného obrázka vidno, že oba implementované algoritmy dosahujú približne rovnakú mieru zhody. Nezáleží preto na použitom type algoritmu, všetky vedú k takmer rovnakým výsledkom. Podobne si môžeme všimnúť, že čím je väčší priestor na doplnenie, tým lepšie výsledky je možné dosiahnuť.



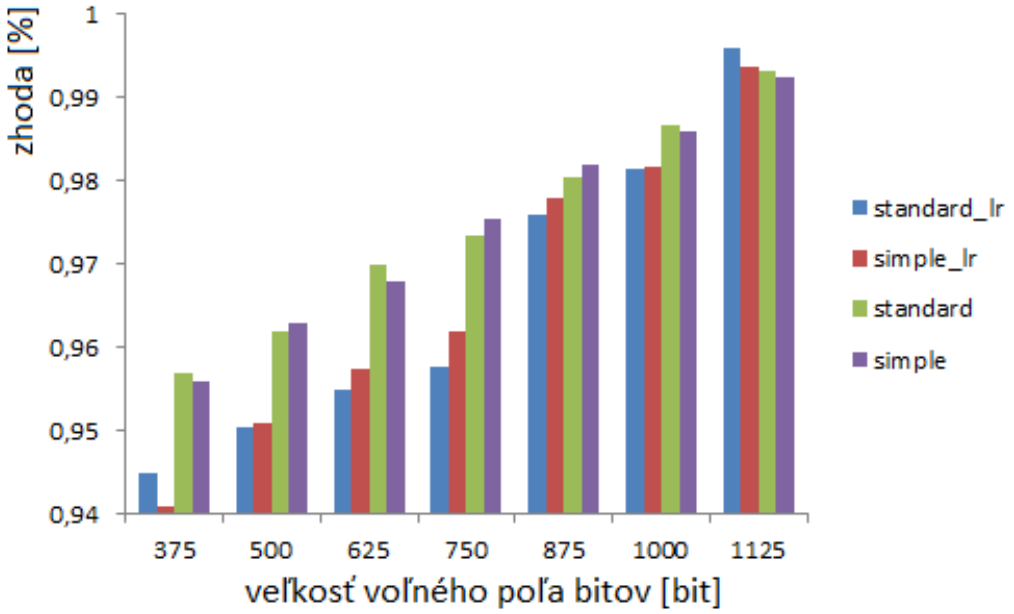
Obr. 8.19: Závislosť miery zhody od veľkosti nosiča a správy pre náhodné plnenie poľa voľných bitov.

Porovnanie časovej náročnosti jednotlivých variácií horolezeckého algoritmu sa nachádza na obrázku č. 8.21. Predpokladali sme, že zjednodušená verzia algoritmu bude rýchlejšia. Ako vidno, pri použití štatistického modulu, ktorý vyhodnocuje mieru zhody na základe bitového porovnávania, je doba výpočtu oboch algoritmov takmer rovnaká. Rozdiel sa ukazuje pri použití štatistického modulu založeného na lineárnej regresii. Tu sa ukazuje, že zjednodušená verzia algoritmu je pri najväčšej veľkosti poľa voľných bitov takmer 50–krát rýchlejšia než štandardná verzia horolezeckého algoritmu.

Po spriemerovaní všetkých ohodnotení jedného aj druhé algoritmu sme zistili, že zjednodušená verzia je vďaka časovej charakteristike približne o 50% výkonnejšia.

Výsledky sú skreslené spôsobom merania času. Merali sme trvanie výpočtu spolu s volaniami ohodnocovacích funkcií. Ako sme mali možnosť vidieť na výslednom grafe, implementácia štatistickej funkcie má výrazný vplyv na výslednú efektivitu. V prípade bitového porovnávania prišlo k extrémnemu skresleniu, keď výhoda zjednodušeného horolezeckého algoritmu bola zanedbateľná vzhľadom na trvanie výpočtu ohodnotenia.

Napokon porovnanie dosiahnutej miery zhody a časovej efektivity všetkých typov algoritmov navzájom sa nachádza na obrázkoch č. 8.22 a 8.23. Nakoľko pre všetky typy algoritmov boli použité dve rôzne nastavenia alebo modifikácie, do výsledného porovnania sme dali priemerné hodnoty každého typu z použitých optimalizačných algoritmov. Pri porovnaní sme použili ohodnocovaciu funkciu štatistického modulu založenej na lineárnej regresii, aby sme eliminovali negatívny vplyv merania časovej efektivity pri bitovom porovnávaní.



Obr. 8.20: Závislosť miery zhody od veľkosti poľa voľných bitov pre horolezecký algoritmus.

`Simple` označuje zjednodušenú verziu algoritmu, `standard` štandardnú (základnú) verziu. Označenie s koncovkou `lr` znamená použitie štatistiky založenej na lineárnej regresii. V opačnom prípade ide o štatistiku založenú na bitovom porovnávaní.

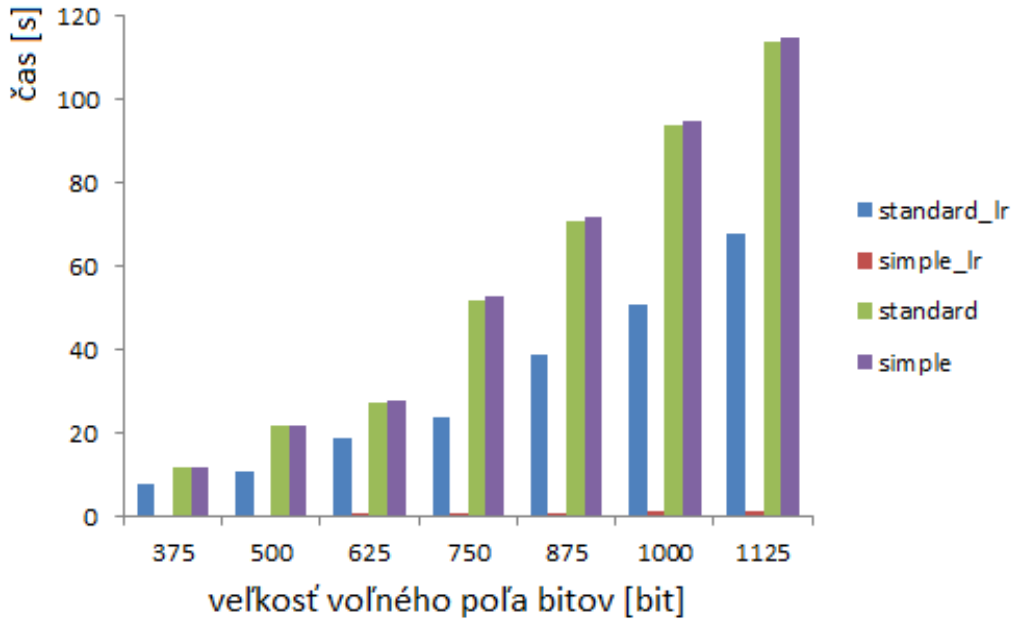
Z grafu vzájomného porovnania dosiahnutej miery zhody vidíme, že najlepším optimalizačným algoritmom je horolezecký algoritmus. S rastúcou veľkosťou poľa voľných bitov dosahuje lepšie výsledky.

Zaujímavým je výsledok genetických algoritmov, ktoré dosahujú najlepšie výsledky pri rozdelení kapacity nosiča medzi správu a voľné pole bitov v pomere 1:1, takže pole voľných bitov zaberá 50% tejto kapacity. S rastúcou veľkosťou tohto poľa nad 50% kapacity radikálne klesá účinnosť genetických algoritmov.

Napokon neurónové siete vykazujú nezávislosť od veľkosti poľa voľných bitov. Aj keď najlepšie výsledky tiež dosiahli pri 50% obsadení kapacity nosiča poľom voľných bitov, ukazuje sa, že celková úspešnosť neurónových sietí je rovnomerná bez ohľadu na veľkosť nosiča a správy.

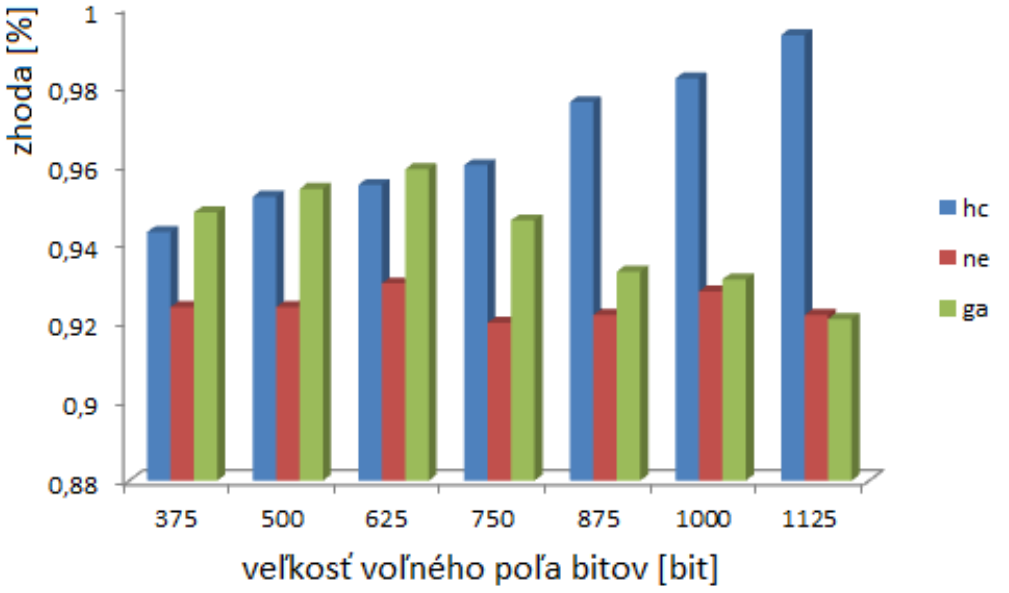
Aj napriek vynikajúcim výsledkom horolezeckého algoritmu vzhľadom na dosiahnutú mieru zhody originálneho a steganografického média pri danej ohodnocovacej štatistike, je v praxi nepoužiteľný, čo naznačuje graf porovnania časovej efektivity (obr. č. 8.23).

Neurónové siete v ňom majú nulovú hodnotu, nakoľko boli o niekoľko rádov rýchlejšie než horolezecký algoritmus. Môžeme si všimnúť, že kým tendencia trvania výpočtu genetických algoritmov je s rastúcim poľom voľných bitov klesajúca, horolezecký algoritmus vykazuje vysoký nárast doby výpočtu.

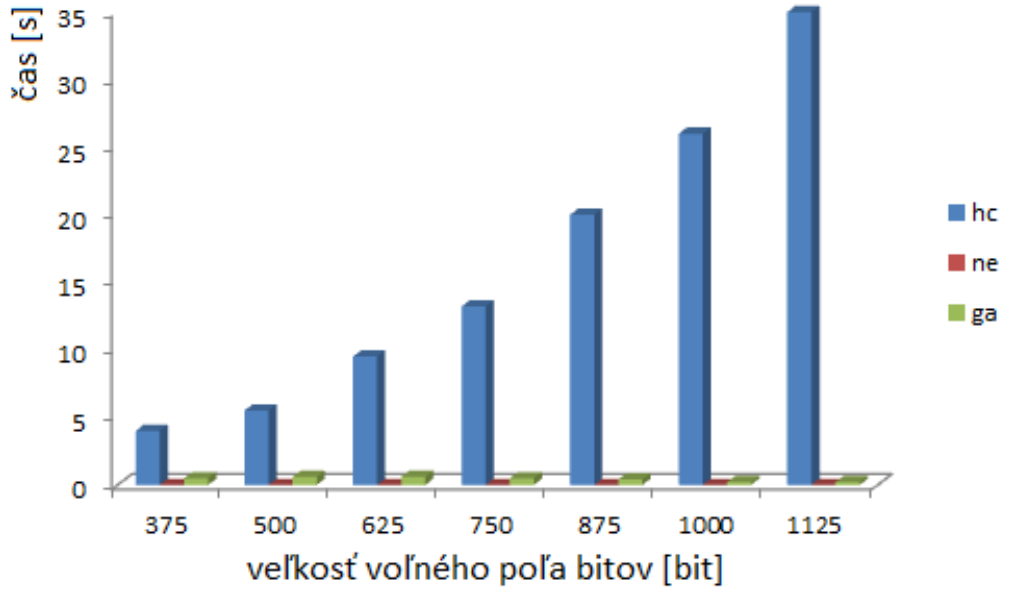


Obr. 8.21: Závislosť trvania výpočtu od veľkosti poľa voľných bitov pre horolezecký algoritmus.

`Simple` označuje zjednodušenú verziu algoritmu, `standard` štandardnú (základnú) verziu. Označenie s koncovkou `lr` znamená použitie štatistiky založenej na lineárnej regresii. V opačnom prípade ide o štatistiku založenú na bitovom porovnávaní.



Obr. 8.22: Vzájomné porovnanie dosiahnutej miery zhody jednotlivých algoritmov. `Ga` označuje genetické algoritmy, `ne` neurónové siete a `hc` horolezecký algoritmus.



Obr. 8.23: Vzájomné porovnanie časovej efektivity jednotlivých algoritmov. `Ga` označuje genetické algoritmy, `ne` neurónové siete a `hc` horolezecký algoritmus.

Genetické algoritmy sú napriek nevýhode dosahovania nižšej miery zhody takmer 20 násobne výkonnejšie než horolezecký algoritmus. Záver: pri optimalizácii štatistických vlastností steganografického nosiča je vhodnejšie používať genetické algoritmy než dopredné neurónové siete s jednou skrytou vrstvou alebo horolezecký algoritmus v jeho základnej verzii.

Záver

Počas obdobia výskumu sme zamerali pozornosť na oblasť obrazovej steganografie či už v statickej alebo dynamickej podobe. Navrhli sme a implementovali niekoľko steganografických systémov, ktorých vývoj neustále pokračuje v súvislosti s projektami pre NBÚ SR. Technickým detailom jednotlivých implementácií sme sa v tejto práci nevenovali, nakoľko sú predmetom utajenia.

Hlavným cieľom dizertačnej práce bol návrh, analýza a implementácia steganografického systému v oblasti pohyblivého obrazu. Na dosiahnutie tohto cieľa sme sa museli hlbšie zaoberať možnosťami steganografických systémov založených na statickom obraze. Ako bolo ukázané v kapitole 2, väčšina steganografických techník pre statické obrázky sa dá použiť aj v oblasti pohyblivého obrazu. Preto sme počas nášho výskumu venovali pozornosť návrhu a analýze steganografických systémov založených na:

- súborovom formáte `BMP` (kapitola 4),
- nekomprimovanom video prúde (kapitola 5),
- súborovom formáte `JPEG` (kapitola 6),
- komprimovanom video prúde uchovávanom v multimediálnom kontajneri `MP4` (kap. 7).

Táto následnosť návrhu a implementácie jednotlivých systémov nebola náhodná. Jednotlivé systémy na seba nadväzujú. Práca s nekomprimovaným videom predpokladala zvládnutie a pochopenie problematiky nekomprimovaných obrazových údajov. Manipulácia s komprimovaným video prúdom kódovaným v štandarde MPEG vyžaduje porozumenie JPEG kompresie. Napokon návrh steganografického systému založenom na komprimovanom videu v sebe zlučuje špecifické vlastnosti časovo závislého sledu obrázkov a zároveň poznatky z oblasti kódovania a stratovej JPEG kompresie.

V niektorých prípadoch (kapitoly 4.3 až 4.7, 5.3 a 5.4) sme poukázali na limity a obmedzenia použitia steganografických systémov založených na modifikácii najmenej významných bitov nosiča. V iných prípadoch sme dôkladnejšie rozobrali štruktúru návrhu (celá kapitola 7) a možnosti použitia steganoanalytických nástrojov na detegovanie použitia istých techník vkladania (kapitoly 6.2 až 6.5).

V oblasti implementácie steganografických systémov založených na videu zatiaľ jestvuje iba niekoľko akademických špekulácií ohľadom možnosti využitia aj tohto typu nosiča na prenos správ. Preto sme sa v našej práci pokúsili ukázať možnosti praktickej realizácie takýchto systémov (kapitoly 5 a 7). Zvlášť v oblasti komprimovaného videa (kapitola 7) sme použili netradičný prístup, keď sme zamerali našu pozornosť na vlastnosti kontajnera uchovávajúceho informácie o samotnom videu; pričom na samotné vkladanie informácie nie je potrebné zasahovať do jeho kódovania.

Medzi doplnkové tézy dizertačnej práce patrí optimalizácia steganografických systémov pomocou techník umelej inteligencie a evolučných výpočtov. Tomu sme sa venovali v kapitole 8. V nej sme sa zaoberali možnosťou optimalizácie steganografických systémov v zmysle zvyšovania odolnosti voči detegovaniu vloženia správy pomocou sledovania istých štatistík (kapitola 8.2) jednotlivých nosičov. Na tento účel sme implementovali model steganografického systému (kapitola 8.1), na ktorom sme realizovali experimentálne merania (kapitola 8.7) úspešnosti genetických algoritmov (kapitola 8.3), neurónových sietí (kapitola 8.4) a horolezeckého algoritmu (kapitola 8.5).

Ukázali sme, že v oblasti vyrovnávania sledovaných štatistík je najvýhodnejšie použiť genetické algoritmy. Použité dopredné neurónové siete s jednou skrytou vrstvou sú vhodnejšie na účely klasifikačných problémov, a horolezecký algoritmus v základnej verzii je z časového hľadiska neefektívny.

Literatúra

[1] J. Čapek and P. Fabian. *Komprimace dát, princípy a praxe.* Computer Press, Brno, 2000. ISBN 80-7226-231-9, pp. 186.

[2] M. Arnold, M. Schmucker, and S. Wolthusen. *Techniques and Applications of Digital Watermarking and Content Protection.* Artech House, London, 2003. ISBN 1-58053-111-3, pp. 274.

[3] I. Avcibas, B. Sankur, and K. Sayood. *Statistical evaluation of image quality measures*, volume 11, No. 2. J. Electron. Imaging, April 2002. pp. 206–223.

[4] J. Baroš. *Miera poškodenia videa po vložení na verejné úložisko anonymného zdieľania multimediálneho obsahu.* Slovenská technická univerzita, Bratislava, 2011. Diplomová práca, pp. 61.

[5] Friedrich L. Bauer. *Kryptology — Methods and Maxims*, volume 149/1983. Lecture Notes in Computer Science, Springer–Verlag, Berlin, January 1995. ISBN 978-3-540-11993-7, pp. 31–46.

[6] W. Bender. *Techniques for Data Hiding*, volume 35, No. 3-4. IBM Corp., Riverton, 1996. IBM Systems Journal, pp. 313–336.

[7] R. J. Berger and B. G. Mobasseri. *Watermarking in JPEG Bitstream.* SPIE Proc. on Security and Watermarking of Multimedia Contents III, San Jose, USA, Jan 2005.

[8] R. Blaško, M. Dubovský, B. Šulej, and T. Marko. *Návrh pomocných modulov steganografického systému.* Slovenská technická univerzita, Bratislava, 2010. Tímový projekt.

[9] U. Budhia, D. Kundur, and T. Zourntos. *Digital Video Steganalysis Exploiting Statistical Visibility in the Temporal Domain*, volume (1)4. IEEE Trans. Information Forensics and security, 2006. pp. 502–516.

[10] Christian Cachin. *An Information-Theoretic Model for Steganography*, volume 1525/1998. Springer, New York, January 1998. Parts of this paper appeared in Proc. 2nd Workshop on Information Hiding. ISBN 978-3-540-65386-8, pp. 306–318.

[11] R. Chandramouli, M. Kharrazi, and N. Memon. *Image steganography and steganalysis: Concepts and practice*, volume 2939/2004. Springer-Verlag, Berlin, February 2004. Digital Watermarking, 2nd International Workshop, IWDW 2003, Seoul, Korea. ISBN 978-3-540-21061-0, pp. 204–211.

[12] B. Chen and G. W Wornell. *Quantization index modulation: a class of provably good methods for digital watermarking and information embedding*, volume 47(4). IEEE Transactions on Information Theory, 2001. pp. 1423—1443.

[13] D. Cinalli, B. G. Mobasseri, and C. O'Connor. *Metadata Embedding in Compressed UAV Video.* Intelligent Ship Symposium, Philadelphia, May 2003.

[14] R. Cinkais. *Detekce steganografie*, volume 6. HAKIN9, 2007. pp. 64–69.

[15] Eric Cole. *Hiding in Plain Sight — Steganography and the Art of Covert Communication.* Wiley John & Sons, New York, 2003. ISBN 0471444499.

[16] I. Cox, J. Fridrich, M. Miller, J. Bloom, and T. Kalker. *Digital watermarking and steganography.* Morgan Kaufmann, Massachusetts, 2008. ISBN 978-0-12-372585-1.

[17] D. L. Currie and C. E. Irvine. *Surmounting the effects of lossy compression on Steganography.* Conference paper, October 1996.

[18] Y. J. Dai, L. H. Zhang, and Y. X. Yang. *A New Method of MPEG Video Watermarking Technology*, volume 2. International Conference on Communication Technology Proceedings (ICCT 2003), April 2003. pp. 11845–1847.

[19] K. A. DeJong and W. M. Spears. *An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms.* Springer-Verlag, Berlin, 1990. Proc. First Workshop Parallel Problem Solving from Nature, pp. 38–47.

[20] DevX. Graphics technical options and decisions. Dostupné elektronicky na `http://www.devx.com.projectcool/Article/19997`, January 2000. Sprístupnené 2. 11. 2009.

[21] Michal Dobeš. *Zpracování obrazu a algoritmy v C#.* Ben, Praha, 2008. ISBN 9788073002336.

[22] S. Dumitrescu, X. L. Wu, and Z. Wang. *Detection of LSB Steganography via Sample Pair Analysis*, volume 2578. Springer-Verlag, Berlin Heidelberg New York, 2002. Petitcolas, F.A.P. (ed.): Information Hiding 5th International Workshop, pp. 355–372.

[23] Nameer N. EL-Emam. *Hiding a Large Amount of Data with High Security Using Steganography Algorithm*, volume 3. Spring, 2007. Journal of Computer Science, ISSN 1549-3636, pp. 223–232.

[24] G. El Loco. A few tools to discover hidden data. Dostupné elektronicky na `http://www.guillermito2.net/stegano/tools/index.html`. Sprístupnené 07. 07. 2011.

[25] G. El Loco. Analyzing steganography softwares. Dostupné elektronicky na `http://www.guillermito2.net/stegano`, 2004. Sprístupnené 04. 04. 2010.

[26] G. C. Langelaar et al. *Watermarking Digital Image and Video Data*, volume 17, No. 5. IEEE Signal Processing Magazine, Sept 2000. pp. 20—46.

[27] H. Farid and L. Siwei. *Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines*, volume 2578. Springer-Verlag, Berlin Heidelberg New York, 2002. Petitcolas, F.A.P. (ed.): Information Hiding 5th International Workshop, pp. 340–354.

[28] P. Fernando and T. Ebrahimi. *The MPEG-4 book*. Upper Saddle River, NJ, Prentice Hall PTR, 2002. ISBN 0-13-061621-4.

[29] J. Fridrich, R. Du, and L. Meng. *Steganalysis of LSB Encoding in Color Images*. Proceedings IEEE International Conference on Multimedia and Expo, New York City, NY, July 30 - August 2 2000. pp. 907–918.

[30] J. Fridrich and M. Goljan. *Practical Steganalysis of Digital Images —- State of the Art*, volume 4675. In Delp III, Security and Watermarking of Multimedia Contents IV, New York City, NY, 2002. pp. 1–13.

[31] J. Fridrich, M. Goljan, and D. Hogea. *Steganography of JPEG images: Breaking the F5 algorithm*. Springer-Verlag, Berlin Heidelberg, 2003. Information Hiding (5th International Workshop 2002), LNCS 2578, pp. 310—323.

[32] Jessica Fridrich. *Feature-Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes*. Springer-Verlag, Berlin Heidelberg, 2004. 6th Information Hiding Workshop, pp. 67–81.

[33] J.J. Grefenstette. *Optimization of Control Parameters for Genetic Algorithms*, volume SMC-16, No. 1. Jan/Feb 1986. IEEE Trans. Systems, Man, and Cybernetics, pp. 122–128.

[34] O. Grošek, M. Vojvoda, and R. Krchnavý. *A New Matrix Test for Randomness*, volume 85. Computing, 2009. ISSN 0010-485X, pp. 21–36.

[35] J. Harmsen and W. Pearlman. *Steganalysis of additive noise modelable information hiding*, volume 5022. Proc. SPIE Security and Watermarking of Multimedia Contents V, Santa Clara, CA, January 2003.

[36] J. J. Harmsen and W. A. Pearlman. *Capacity of steganalysis channels*. Proceedings of the 7th Workshop on Multimedia and Security, New York, 2005. ISBN 1-59593-032-9, pp. 11–24.

[37] F. Hartung and B. Girod. *Watermarking of uncompressed and compressed video*, volume 66(3). Signal Processing, Special Issue on Copyright Protection and Access Control for Multimedia Services, 1998. pp. 283–301.

[38] S. Haykin. *Neural Networks – A Comprehensive Foundation*. Prentice Hall, New Jersey, 2 edition, 1998. ISBN 978-0132733502.

[39] R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley Publishing Company, Massachusetts, 1990. ISBN 978-0201093551.

[40] Josef Hynek. *Genetické algoritmy a genetické programování.* Grada Publishing, Praha, 2008. ISBN 978-80-247-2695-3, pp. 200.

[41] J. S. Jainsky, D. Kundur, and D. R. Halverson. *Towards Digital Video Steganalysis using Asymptotic Memoryless Detection*, volume (1)4. MM&Sec'07, Dallas, Texas, USA, September 20-21 2007.

[42] M. Jókay. *The design of a steganographic system based on the internal MP4 file structures.* International Journal of Computers and Communications, 2011. Preprint, pp. 8.

[43] M. Jókay and J. Baroš. *On the suitability of the internet multimedia storage for steganographic information transfer in MP4 files.* Kybernetika, 2011. Preprint, pp. 14.

[44] N. F. Johnson and S. Jajodia. *Steganalysis of Images Created Using Current Steganography Software*, volume 1525. Springer-Verlag, Oregon, April 1998. Proceedings of the 2nd International Workshop on Information Hiding, ISBN 3-540-65386-4, pp. 273–289.

[45] S. Katzenbeisser and Fabien Petitcolas. *On defining security in steganographic systems*, volume 4675. Society of Photo-Optical Instrumentation Engineers, Bellingham, January 2002. SPIE proceeding series. ISBN 0-8194-4415-4, pp. 50–56.

[46] M. Kharrazi, H. T. Sencar, and N. Memon. Cover selection for steganographic embedding. Dostupné elektronicky na `http://sharif.edu/ kharrazi/pubs/icip06.pdf`, 2006. Sprístupnené 28. 10. 2009.

[47] Greg Kipper. *Investigator's Guide to Steganography.* Auerbach Publications, Boka Raton, 2004. ISBN 0849324335.

[48] V. Kvasnička. *Úvod do teórie neurónových sietí.* Iris, Bratislava, 1997. ISBN 80-88778-30-1.

[49] V. Kvasnička, J. Pospíchal, and P. Tiňo. *Evolučné algoritmy.* Vydavateľstvo STU, Bratislava, 2000. ISBN 80-227-1377-5.

[50] G. C. Langelaar and R. L. Lagendijk. *Optimal Differential Energy Watermarking of DCT Encoded Images and Video*, volume 10(1). IEEE Transactions on Image Processing, 2001. pp. 148–158.

[51] B. Liu, F. Liu, Ch. Yang, and Y. Sun. *Secure Steganography in Compressed Video Bitstreams.* The Third International Conference on Availabity, Reliability and Security (ARES 08), March 4-7 2008. DOI 10.1109/ARES.2008.140. ISBN 978-0-7695-3102-1, pp. 1382–1387.

[52] H. H. Liu and L. W. Chang. *Realtime digital video watermarking for digital rights management via modification of VLCs.* Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS'05), 2005.

[53] X. Liu, F. Liu, B. Lu, and X. Luo. *Real-Time Steganography in Compressed Video.* Proceedings of MRCS 2006, LNCS 4105, Springer-Verlag, Aug 2006. pp. 43–48.

[54] C. S. Lu, J. R. Chen, and K. C. Fan. *Real-time framedependent video watermarking in VLC domain*, volume 20. Signal Processing: Image Communication, 2005. pp. 624–642.

[55] B. G. Mobasseri and M. P. Marcinak. *Watermarking of MPEG-2 Video in Compressed Domain Using VLC Mapping.* ACM Multimedia and Security Workshop 2005, New York, NY, Aug 2005.

[56] T. Moravčík. *Steganografia v obrazovom súbore JPG.* Slovenská technická univerzita, Bratislava, 2010. Diplomová práca.

[57] J. D. Murray and W. van Ryper. *Encyklopedie grafických formátøu*, volume 6. Computer Press, druhé vydanie edition, 1997. ISBN 80-7226-033-2.

[58] Nvidia. Developer's zone. Dostupné elektronicky na `http://http://developer.nvidia.com/category/zone/cuda-zone`, 2011. Sprístupnené 11. 07. 2011.

[59] Jan Ondruš. *Bezstrátová komprese JPEG grafiky.* Karlova Univerzita, Praha, 2009. Diplomová práca, pp. 40.

[60] M. Oravec, J. Polec, and S. Marchevský. *Neurónové siete pre číslicové spracovanie signálov.* Faber, Bratislava, 1998. ISBN 80-967503-9-9.

[61] Fernando Perez-Gonzales and Juan Hernandez. A tutorial on digital watermarking. Dostupné elektronicky na `http://www.gts.tsc.uvigo.es/gpsc/publications/wmark/carnahan99.pdf`. Sprístupnené 11. 05. 2009.

[62] Fabien Petitcolas. Stirmark benchmark 4.0. Dostupné elektronicky na `http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark`, 20. September 2008. Sprístupnené 20. 05. 2009.

[63] J. Polec, J. Pavlovičová, and T. Karlubíková. *Medzinárodné štandardy pre kompresiu obrazu II.* Slovenská technická univerzita, Bratislava, 2001. ISBN 80-227-1784-3.

[64] B. Rainer. *Advanced statistical steganalysis.* Springer-Verlag, Berlin, 2010. ISBN 978-3-642-14312-0, pp. 285.

[65] Radovan Ridzoň. *Invariantné metódy digitálnej vodotlače v statických obrazoch.* Dizertačná práca, Technická univerzita, Košice, 2007. Odbor Telekomunikácie.

[66] P. Šrámek. *Steganografický systém s použitím hill-climbing algoritmov.* Slovenská technická univerzita, Bratislava, 2011. Záverečná práca.

[67] N. Rougier. Biological neuron schema. Dostupné elektronicky na `https://commons.wikimedia.org/wiki/File:Neuron-figure.svg`, June 2007. Sprístupnené 13. 8. 2011.

[68] Martin Rusnák. Regresná analýza. Dostupné elektronicky na `http://www.healthnet.sk/texts/statistika/regresia/regresia.htm`. Sprístupnené 21. 06. 2011.

[69] P. Sallee. *Model-based steganography.* Springer-Verlag, Berlin Heidelberg, 2004. International Workshop on Digital Watermarking (IWDW 2003), LNCS 2939, pp. 154—167.

[70] A. Sarkar, U. Madhow, S. Chandrasekaran, and B. S. Manjunath. *Adaptive MPEG-2 Video Data Hiding Scheme*, volume 6505. Security, steganography, and watermarking of multimedia contents No9, San Jose CA, 2007. pp. 65051D.1–65051D.9.

[71] Caspar Schott. *Schola steganographia*. Jobus Hertz, Norimberg, 1665.

[72] Ivan Sekaj. Genetické algoritmy — prednášky. Dostupné elektronicky na `http://www.kasr.elf.stuba.sk/index.php?go=studij_mat`, 2010. Sprístupnené 21. 05. 2010.

[73] XRCE TeXnology Showroom. Digital watermarking in printed images. Dostupné elektronicky na `http://www.xrce.xerox.com/showroom/fs/stochas.pdf`. Sprístupnené 11. 05. 2009.

[74] MSSG: Free MPEG software. Chapter on rate control and quantization control. Dostupné elektronicky na `http://www.mpeg.org/MPEG/MSSG/tm5/Ch10/Ch10.html`. Sprístupnené 3. 3. 2011.

[75] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. *Robust image-adaptive data hiding based on erasure and error correction*, volume 13. IEEE Trans. on Image Processing, December 2004. pp. 1627–1639.

[76] D. Sovič. Kompresný štandard jpeg. Dostupné elektronicky na `http://www.pakuj.host.sk/jpeg/jpeg.html`, 2009. Sprístupnené 04. 04. 2010.

[77] K. Su, D. Kundur, and D. Hatzinakos. *Statistical invisibility for collusionresistant digital video watermarking*, volume (7)1. IEEE Trans. Multimedia, 2005. pp. 43–51.

[78] Ioannes Trithemius. *Polygraphiae libri sex*. Ioannes Birckmannus & Theodorus Baumius, Cologne, 1586.

[79] Gregory K. Wallace. *The JPEG Still Picture Compression Standard*, volume 34. ACM, New York, April 1991. Communications of the ACM, ISSN 0001-0782, pp. 30–44.

[80] A. Westfeld. *F5–A Steganographic Algorithm*, volume 2137. Proceedings of 4th International Conference on Informatio Hiding, Springer-Verlag, 2001. pp. 289–302.

[81] A. Westfeld and A. Pfitzmann. *Attacks on Steganographic Systems*. Springer-Verlag, Berlin Heidelberg, 1999. Information Hiding, Third International Workshop, IH'99, Dresden, Germany, September/October, 1999, Proceedings, LNCS 1768, pp. 61–76.

[82] A. Westfeld and A. Pfitzmann. *Attacks on Steganographic Systems*, volume 1768. Springer-Verlag, Berlin, 2000. pp. 61–75.

[83] C. Xu, X. Ping, and T. Zhang. *Steganography in Compressed Video Stream*. Proceedings of the First International Conference of Innovative Computing, Information and Control, 2006. ISBN 0-7695-2616-0.

[84] D. P. Ye, C. F. Zou, Y. W. Dai, and Z. Q. Wang. *New adaptive watermarking for real-time MPEG videos*, volume 185. Applied Mathematics and Computation, 2007. pp. 907–918.

[85] Zuzana Zeťáková. Genetické algoritmy i. Dostupné elektronicky na `http://neuron-ai.tuke.sk/ãrvayova/bk`. Sprístupnené 23. 05. 2010.

[86] J. Zhang, J. Li, and L. Zhang. *Video Watermark Technique in Motion Vector*. Proceedings of XIV Brazilian Symposium on Computer Graphics and Image Processing, 2001. pp. 179–182.

[87] W. Zhang, X. Zhang, and S. Wang. *Maximizing Steganographic Embedding Efficiency by Combining Hamming Codes and Wet Paper Codes*. Springer-Verlag, 2008. Information Hiding, 10th International Workshop, pp. 60–71.

Register