VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

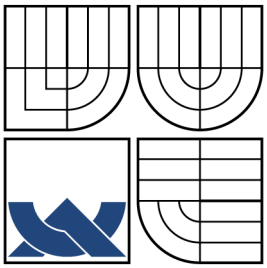
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

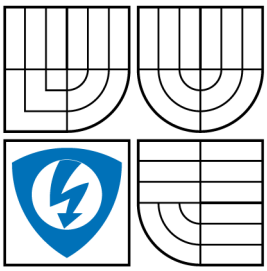
LABORATORNÍ ÚLOHA SKRÝVÁNÍ DAT VE ZVUKOVÉM ZÁZNAMU

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. JIŘÍ KORTUS
AUTHOR



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

LABORATORNÍ ÚLOHA SKRÝVÁNÍ DAT VE ZVUKOVÉM ZÁZNAMU

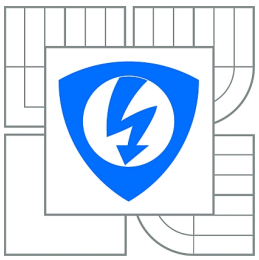
LABORATORY EXERCISE IN DATA HIDING IN THE AUDIO RECORD

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR
Bc. JIŘÍ KORTUS

VEDOUCÍ PRÁCE
SUPERVISOR
doc. Ing. KAREL BURDA, CSc.

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Jiří Kortus	***ID:***	126760
Ročník:	2	***Akademický rok:***	2013/2014

NÁZEV TÉMATU:

Laboratorní úloha skrývání dat ve zvukovém záznamu

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte a popište problematiku skrývání dat ve zvukových záznamech. Na tomto základě navrhněte a zrealizujte počítačový výukový program ke skrývání dat ve zvukových souborech. Pro vytvořený program navrhněte laboratorní úlohu, v jejímž rámci by si studenti mohli ověřit vliv hustoty skrývaných dat na bezpečnost skrytí dat a vliv kryptografických ochran na bezpečnost skrývaných dat. Pro laboratorní úlohu vypracujte návod a potřebnou dokumentaci.

DOPORUČENÁ LITERATURA:

[1] Žilka, R.: Steganografie a stegoanalýza. Masarykova univerzita, Brno 2008.
[2] Morkus, F.: Program pro skrývání dat v obrazových souborech. VUT v Brně, Brno 2011.

Termín zadání:	10.2.2014	***Termín odevzdání:***	28.5.2014

Vedoucí práce: doc. Ing. Karel Burda, CSc.
Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Diplomová práce se věnuje problematice skrývání dat do zvukových záznamů. Popisuje obecné steganografické principy a dále se zaměřuje na specifika skrývání dat ve zvukových záznamech a rozebírá steganografickou techniku LSB a vhodnost různých typů zvukových záznamů pro skrývání dat touto technikou. Na základě popsaných mechanismů je navržena laboratorní úloha zabývající se skrýváním dat do zvukových záznamů a vlivem některých aspektů na výsledný záznam nesoucí skrytá data. Dále je popsán program, který byl vytvořen jako součást laboratorní úlohy, a to z hlediska návrhového, funkčního i implementačního. V rámci laboratorní úlohy si budou studenti moci vyzkoušet vliv množství skrývaných dat a různých způsobů jejich skrytí na kvalitu výsledného zvukového záznamu, a na základě toho si vytvořit bližší představu o problematice skrývání dat do zvukových záznamů technikou LSB.

KLÍČOVÁ SLOVA

skrývání dat steganografie LSB WAV laboratorní úloha Python

ABSTRACT

The diploma thesis aims on the matter of data hiding (steganography) in audio records. It describes general steganographic principles and aims further on the specifics of data hiding in audio records and also aims on the LSB steganographic technique and suitability of different types of audio records to be used with this technique. The thesis also describes a laboratory exercise focused on steganography in audio records and influence of related aspects on the final audio record which contains secret data. Further, the thesis describes a program that was created especially for the laboratory exercise, from the design, functional as well as implementation-related view. Within the exercise, students will be able to examine how the amount of data to be hidden in the audio record and different ways of data hiding will affect quality of the resulting audio record, and therefore they can become more familiar with the matter of steganography based on the LSB method used in audio records .

KEYWORDS

data hiding steganography LSB WAV laboratory exercise Python

KORTUS, Jiří *Laboratorní úloha Skrývání dat ve zvukovém záznamu*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 71 s. Vedoucí práce byl doc. Ing. Karel Burda, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Laboratorní úloha Skrývání dat ve zvukovém záznamu“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno ....

(podpis autora)

Poděkování

Rád bych poděkoval panu doc. Ing. Karlu Burdovi, CSc., za konzultaci a rady související s touto prací.

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Lidskou komunikaci již od dávných dob kromě problému, jak informaci předat, provází nezřídka další související problém, a to jak informaci na její cestě od odesílatele k příjemci utajit. Během času bylo vynalezeno mnoho metod pro předávání či ukládání tajných zpráv v rámci jiných zpráv nebo i fyzických objektů. První metody byly velmi jednoduché, avšak s historickým vývojem se neustále zdokonalovaly, byly sofistikovanější a snižovaly riziko odhalení nebo dokonce zajišťovaly důvěrnost dotyčných zpráv. Vývoj vyústil ve vznik samostatné disciplíny, která se utajováním informací zabývá – steganografie.

Vymezení steganografie mělo za následek určení formálního rámce v oblasti utajování dat a vedle často související oblasti – kryptografie – se steganografie rozvíjela a rozvíjí zejména v souvislosti s příchodem informačních technologií. Ty kladou nové, dříve neznámé či nedostupné možnosti a výzvy nejen z hlediska utajování informací, ale z druhé strany i z hlediska detekce a odhalování utajených informací.

Cílem této práce je popsat obecné steganografické principy a dále se soustředit na možnosti skrývání tajných dat do zvukových záznamů. Na základě těchto poznatků je dále představena navržená laboratorní úloha, která studentům umožní lépe se seznámit s ukrýváním tajných dat do zvukových záznamů. Úloha využívá počítačového programu Stego, který byl pro ni navržen.

První kapitola se věnuje obecnému úvodu do problematiky steganografie a stručnému historickému vývoji. Druhá kapitola rozebírá steganografické principy podrobněji, ve třetí kapitole jsou rozebrány některé metody použitelné pro ukrývání dat ve zvukových záznamech. Kapitola dále do větší míry přibližuje steganografickou techniku LSB, diskutuje vhodnost různých typů a formátů zvukových záznamů pro skrývání dat a popisuje strukturu zvukových souborů WAV. Ve čtvrté kapitole je diskutována laboratorní úloha zabývající se ukrýváním dat ve zvukových záznamech, v níž si studenti mohou vyzkoušet, jaký vliv na zvukový záznam bude mít ukrytí dat na základě množství dat a nastavení jednotlivých parametrů, a dále jak se projeví šifrování dat. Pátá kapitola popisuje program Stego z hlediska návrhového i implementačního a popisuje principy, na kterých je program postaven.

Příloha A obsahuje uživatelskou příručku k programu, která podrobně popisuje možnosti jeho využití. Příloha B obsahuje zadání laboratorní úlohy a poznámky pro vedoucí cvičení.

1 ÚVOD DO STEGANOGRAFIE

1.1 Přiblížení pojmu steganografie

Pojem steganografie [1] pochází z řeckých výrazů steganós (skrytý) a grafein (psát) a označuje vědní obor, který se zabývá utajením informace v rámci jiné, neutajené zprávy (nebo obecněji i fyzického objektu). Přesněji – jedná se o utajení netriviální, tj. takové, které není snadno odhalitelné (ať již lidskému pozorovateli či stroji) a je provedeno (alespoň do určité míry) sofistikovanou metodou.

Názorným příkladem může být využití „neviditelného“ inkoustu k napsání tajné zprávy buď na čistý list papíru nebo jako tajný dodatek k jinak zcela běžně vyhlížející písemnosti. Z pohledu běžného čtenáře bude daný list papíru obsahovat pouze regulérně viditelný text, zatímco čtenář, který bude mít znalost toho, že kromě viditelného textu je na listu obsažena i skrytá zpráva, může použít prostředek k jejímu zviditelnění (zahřátí papíru, aplikace chemické látky, která způsobí zabarvení neviditelného textu apod.).

Z tohoto příkladu je zcela zřejmých několik faktů, které je nutné brát při použití steganografie (obecně) v úvahu: Příjemce i odesílatel musí znát metodu, kterou je informace skryta, příjemce navíc musí vědět, kde a jakým způsobem má hledat skrytou zprávu. Dále musí brát v úvahu to, že skrytou zprávu mohou odhalit i třetí osoby, což je nežádoucí jev. Může se jednat jak o náhodné zjištění (např. náhodné zahřátí papíru), tak i o zjištění cílené – třetí osoba, např. doručovatel, má podezření, že probíhá tajná komunikace, a může tedy vyvíjet úsilí (za použití jak náhodných, tak i cílených metod) pro zjištění, zda je přenášena tajná zpráva, a dále se může i pokoušet o její přečtení. Z toho vyplývají další dva aspekty spojené se steganografií – rozdíl mezi ukrytím zprávy a zachováním důvěrnosti jejího obsahu. To již však zasahuje do oboru kryptografie, který se ovšem v praxi nezřídka s využitím steganografie v netriviální míře protíná. [2]

1.2 Historie

Zatímco dnes je steganografie považována za obor související s informatikou nebo matematikou [2], ještě před několika desítkami let nebyla jako taková vyčleňována, a postupy, které dnes řadíme do oblasti steganografie, byly jednoduše považovány za více či méně chytré nápady k ukrytí a přenosu informace od odesílatele k příjemci, aniž by bylo vynakládáno zvláštní úsilí pro jejich bližší formální popis a zkoumání.

Nejstarší zmínka o využití steganografie pochází již ze starověku[2] – jednalo se o situaci, ve které bývalý král řecké Sparty, Demaratos, skryl tajnou informaci

o chystaném útoku Peršanů na Spartu na voskovanou dřevěnou destičku. V té době se tyto destičky používaly na psaní (vyrytí textu na destičku), přičemž destička byla vyrobena z hladkého kusu dřeva a následně namočena do roztaveného vosku. V případě destičky s tajnou zprávou Demaratos údajně nejdříve vosk z destičky odstranil, vyryl tajnou zprávu přímo do destičky a poté ji opět pokryl roztaveným voskem, takže vypadala jako kterákoliv jiná voskovaná destička, a následně obchodní cestou doputovala do Sparty.

Další, z dnešního pohledu primitivní, postup k doručení tajných zpráv ve starověkém Řecku zahrnoval uschování zprávy v břiše zajíce. Z oblasti starověké Číny pochází další údajně použitá metoda – napsání zprávy na kus hedvábí, které bylo poté poskládáno a navoskováno. Posel tento kousek hedvábí spolkl a přepravoval jej ve svém trávicím traktu.

Jiná zmínka o steganografii se datuje do prvního století našeho letopočtu [2], kdy filozof Plinius Starší připravil látku, kterou bychom dnes nejspíše označili jako neviditelný inkoust.

Z doby raného novověku pochází tzv. Cardanova mřížka (pojmenovaná po Gerolamu Cardanovi), což byla destička z pevného materiálu obsahující pravoúhlé otvory. Do otvorů v destičce byla po přiložení na list papíru vepsána tajná zpráva a zbylé místo okolo vepsaných písmen bylo doplněno dalším textem. Je zřejmé, že v závislosti na povaze doplněného textu (a tudíž i na celkově sestaveném textu tvořeném písmeny doplněného textu i původní zprávy) mohl výsledný text budit jisté podezření; dalším rysem této metody je s určitou nadsázkou i přesah do oblasti kryptografie – konkrétní podoba Cardanovy mřížky použitá ke skrytí textu by se patrně dala považovat v určitém smyslu za šifrovací klíč.

Další rozvoj zaznamenala oblast steganografie v moderní době, především v souvislosti s první a druhou světovou válkou a potřebou tajné výměny informací bez intervence nepřátelské strany. Mezi steganografické metody z období první světové války patřilo využívání synonym z určité terminologické oblasti pro označování pojmů z jiné oblasti [2] (například na první pohled ničím příliš neobvyklý text obsahoval popis s chemickým či drogistickým názvoslovím nebo fingované obchodní sdělení; ve skutečnosti jednotlivé termíny označovaly územní útvary (města, regiony) nebo nepřátelské válečné jednotky – lodě, letadla apod. a umožňovaly tak předávání strategických válečných informací o postupu nepřítele). V principu se jednalo o podobný přístup jako při použití *argotu*[3], mluvy typické zejména pro kriminálníky, kteří používáním smluvených výrazů před neznalými osobami maskují skutečný význam vyřčených sdělení.

Skrývání tajné zprávy v textu využívala i další metoda: Text obsahující tajné sdělení je sestaven tak, aby na první pohled alespoň zhruba dával smysl; zároveň jsou slova v textu volena takovým způsobem, aby jednotlivá písmena tajného textu

(po řadě) odpovídala počátečním písmenům slov sestavené zprávy. Například věta „**A**lbert **h**ladově **o**kusuje **j**ablko.“ by na základě výše uvedeného postupu měla v sobě ukrytou zprávu „Ahoj“.

Technicky zajímavým pokrokem byl objev tzv. mikroteček nacistickou armádou. Mikrotečka byla miniaturní část fotografického filmu upraveného do tvaru a rozměru odpovídajícímu tečce napsané na list papíru psacím strojem, přičemž na tuto miniaturní plochu bylo možné vměstnat přibližně jednu stránku A4 textu. Mikrotečky byly umísťovány na listy papíru s psaným textem např. na místa teček ve větách, takže na první pohled nebudily pozornost.

S mohutným rozvojem elektroniky ve druhé polovině 20. století, z nějž mimo jiné plynul i rozvoj výpočetní techniky a Internetu, přišly zcela nové možnosti i v oblasti steganografie. Příkladem může být ukrývání dat v komunikačních protokolech, souborových systémech (respektive obecně v diskových oddílech) nebo v obrazových či zvukových záznamech.

2 PRINCIPY STEGANOGRAFIE

Jak již bylo naznačeno, účelem steganografie je ukrytí tajné informace (tajné zprávy, například strategických informací vojenského charakteru) uvnitř jistého prostředku, který je do určité míry přístupný třetím stranám, tedy do tzv. *nosiče*. V kontextu informačních technologií mohou být jako nosiče využity mj. tyto prostředky[2][1]:

- spustitelné soubory,
- obrazové, zvukové či multimediální soubory,
- datové jednotky komunikačních (síťových) protokolů,
- diskové oddíly, souborové systémy,
- textové soubory,
- nevyžádaná pošta (spam)

Výslednou entitu, která vznikne kombinací nosiče a vložené tajné zprávy, nazýváme *steganogramem*:

$$\text{nosič} + \text{tajná zpráva} = \text{steganogram}$$

Pod pojmem *tajná zpráva* je chápána informace prakticky libovolného charakteru – nemusí se jednat doslovně o zprávu v běžném slova smyslu (textové sdělení), ale může se jednat kupříkladu o obrazová data, zvukový záznam, logickou hodnotu, nebo jakoukoliv jinou reprezentaci požadované informace.

Při ukrývání tajné informace do nosiče lze postupovat dvěma způsoby: První možností je přidat tajnou zprávu do nosiče ve formě redundantních dat (pokud to struktura nosiče dovoluje, tj. pokud lze provést přidání takovým způsobem, aby původní sémantika ani obsah nosiče nebyl poškozen). To je možné provést například u spustitelných souborů nebo u jiných nosičů, které dovolují dynamicky upravovat obsah, a to takovým způsobem, že při běžném použití nosiče nebudou data reprezentující ukrytou zprávu nijak interpretována, nebo alespoň bude pravděpodobnost interpretace dat dostatečně nízká. Výhodou redundantního přidání tajných dat do nosiče je, že vlastní informace obsažená v samotném nosiči není ovlivněna nebo poškozena; nevýhodou je, že přidaná data mohou být při analýze steganogramu odhalena, nebo může steganogram ve srovnání s běžným nosičem stejného typu po úpravě budit podezření (změnou velikosti souboru, obsahem segmentů, které nemají jakoukoliv faktickou spojitost s ostatními částmi nosiče, apod.).

Druhou možností je modifikovat část nosiče (myšleno část jakožto souhrn více podčástí celku, nikoliv nutně souvislý úsek dat nosiče) tak, že určitý podíl původních dat v nosiči se nahradí daty tajné zprávy. Tento postup může být, v závislosti na použitém nosiči, poměrně jednoduchý na implementaci. Je však vhodný pouze pro takové nosiče, jejichž modifikace (*v rozumné míře*) nezpůsobí příliš velké změny ve

statistických vlastnostech dat v nosiči nebo nebudou znatelné z hlediska smyslových vjemů pozorovatele (zrak v případě obrazového nosiče, sluch v případě zvukového nosiče) apod. To, co lze označit za rozumnou míru, se může velmi podstatně lišit dle použitého nosiče, dle konkrétního návrhu a implementace steganografické metody, množství ukryté informace, *steganografické kapacity nosiče*[1] a případně dalších souvislostí.

2.1 Stegoanalýza

Na ukrývání dat lze pohlížet z hlediska tvůrce *stegosystému* (systému, který zajišťuje zejména ukrytí tajné zprávy, přenos nebo uchování steganogramu a extrakci tajné zprávy příjemcem). Tvůrce stegosystému se zabývá volbou vhodného nosiče, volbou způsobu skrytí dat do nosiče, dalšími technickými náležitostmi a v neposlední řadě též zhodnocením, jaký dopad by mělo případné odhalení existence tajné zprávy nebo jejího obsahu.

Opačný pohled na steganografii, respektive konkrétní stegosystém, se skýtá útočníkovi, tedy třetí straně, u které je z hlediska odesílatele a příjemce staganogramu nežádoucí, aby byla schopná obsah utajené zprávy či pouze informaci o její existenci odhalit. Zde se dostáváme do oblasti *stegoanalýzy*[4], tedy oboru, jehož účelem je analyzovat zachycenou zprávu a stanovit s určitou pravděpodobností, zda neobsahuje utajenou informaci, a případně tuto utajenou informaci dále extrahovat.

Stegoanalýza může být prováděna strojově na základě zkoumání statistických vlastností či analýzy vybraných jevů v rámci dané zprávy, které lze kvantitativně vyjádřit[2], a dále také porovnáním zjištěných skutečností s profilem podobných typů zpráv (na základě znalosti používaných stegosystémů a jimi vytvořených steganogramů). Zde se jedná o tzv. *statistický útok*. *Útok* v tomto kontextu znamená aktivitu, jejímž cílem je odhalit tajnou zprávu ve steganogramu nebo dokonce konkrétního původního nosiče bez ukryté zprávy. Znepřístupnění nosiče v původní podobě (bez skryté zprávy) třetím stranám je dalším z důležitých aspektů steganografie, které je vhodné brát v úvahu při návrhu nebo používání určitého stegosystému; v opačném případě může mít potenciální útočník podstatně usnadněnou práci při analýze steganogramu.

Další možností strojové stegoanalýzy je *strukturální útok*[2], při kterém je zkoumána struktura daného nosiče, neobvykle použité části (segmenty) uvnitř nosiče,

[1]Steganografickou kapacitou rozumíme množství informace, které lze do nosiče ukrýt. Podle použitého způsobu závisí buď na zvolené míře redundance (množství dat vložených do nosiče) nebo na zvolené míře nahrazení původních dat v nosiči tajnou zprávou. To vše platí při splnění podmínky, že dané množství tajných dat neohrozí utajení přítomnosti tajné zprávy, nebo přímo obsah samotné zprávy.

nadbytečná/nereferencovaná data aj. V rámci stegoanalýzy může být útok na steganogram prováděn i člověkem, což se týká zejména použití obrazových či zvukových záznamů coby nosičů pro ukrytí tajné zprávy. V tomto případě pozorovatel zkoumá daný steganogram a snaží se v něm najít nějaké nesrovnalosti (ve spojení s jeho znalostí jiných běžně se vyskytujících obrazů či zvuků). Příkladem mohou být neobvyklé artefakty v obrazu, poškození barevné reprezentace jednotlivých bodů, zvýšená hladina šumu ve zvukovém záznamu nebo jiné jevy, které se obvykle v daném typu nosičů ve větší míře nevyskytují.

Pokud je cílem chránit utajené zprávy před útokem prováděným člověkem (nebo, jinak řečeno, nechceme, aby si pozorovatel, který náhodně či cíleně přijde se steganogramem do styku, všimnul čehokoliv neobvyklého), můžeme využít prozkoumaných nedostatků lidských smyslů. Nejjednodušší možností, jak toho dosáhnout, je snížení míry vkládané tajné informace do nosiče (čímž dojde jen k takovému ovlivnění, u kterého se dá předpokládat, že si jej pozorovatel nevšimne) – například při použití techniky ukrývání dat *LSB* (blíže popsána v kapitole 3.2).

Jinou možností je využití znalostí o psychovizuálním modelu, tedy popisu toho, jak člověk vnímá obraz. V této oblasti se jedná především o citlivost lidského zraku na změny barevné a jasové složky (změny v jasové složce jsou pro zrak markantnější než změny v barevné složce, čehož se mj. využívá například ve ztrátových kompresích, typicky ve formátu JPEG). Analogicky lze u zvukových záznamů využít znalosti psychoakustického modelu (tj. jak lidský sluch vnímá zvuky, těchto znalostí je opět využito mj. ve ztrátových kompresních formátech), a pro ukrytí tajné zprávy využít např. jevů známých jako *maskování* v *časové* nebo *kmitočtové* oblasti (princip je popsán v kap. 3).

3 STEGANOGRAFIE VE ZVUKOVÝCH ZÁZNAMECH

Vkládání tajných dat do zvukových záznamů lze založit buď na technice nahrazení nejméně významných bitů – *LSB* (popsána dále), nebo využít nedokonalosti lidského sluchu a data vložit jiným způsobem než prostým nahrazením bitů ve vzorcích.[1] Uveďme alespoň některé ze steganografických technik, které lze ve zvukových záznamech použít:

3.1 Využívané techniky

Maskování v časové oblasti. V tomto případě se využívá faktu, že lidský sluch nedokáže zachytit krátký zvuk o určité intenzitě (označován jako *maskovaný*), který předchází nebo následuje ve velmi krátké době (přibližně desítky milisekund) po jiném zvuku o podstatně vyšší intenzitě (hladině akustického tlaku), označovaném jako *maskovací*. To, jaké zvuky budou maskovány, je vyjádřeno tzv. *maskovacím prahem*. Do zvukového záznamu je tedy možné vložit tajná data, která budou reprezentována zvuky o nízké intenzitě časově umístěné v záznamu tak, aby se ukryly pod maskovací práh zvuků s vyšší intenzitou.

Někdy se též uvádí jako jedna z technik **vkládání ozvěny** [5], což je de facto technika spadající do maskování v časové oblasti. Konkrétním kritériem dle [5] je, že vložená ozvěna musí mít zpoždění do 2 ms a její intenzita musí být menší než intenzita zvuku, po kterém následuje. Ukrývaná zpráva tak může být kódována např. přítomností nebo absencí ozvěny v daných místech zvukového záznamu.

Maskování v kmitočtové oblasti. Jedná se techniku založenou na podobném principu jako v předchozím případě, avšak v tomto případě se jedná o využití maskování v kmitočtové oblasti. Vychází se opět z vlastností lidského sluchu, konkrétně jeho neschopnosti zachytit zvuk (maskovaný) o určité intenzitě (s určitou hladinou akustického tlaku), jehož kmitočet se ve spektru nachází nedaleko jiného zvuku (maskovacího) o nepříliš rozdílném kmitočtu a s vyšší intenzitou.

Změna fáze. Sluch není citlivý na absolutní změnu fáze ve zvukovém signálu, čehož lze také využít pro uložení tajných dat [5]. Princip spočívá v rozdělení signálu

[1] I u zvukových záznamů je možné vložit – pokud to umožňuje daný formát zvukového souboru – tajná data tak, aby nebyl ovlivněn zvukový záznam, např. přidáním nového neinterpretovaného datového segmentu, nicméně takto vložená data by pravděpodobně i při základní analýze působila hned na první pohled podezřele.

na fragmenty (jejichž délka a pozice musí být známá i při extrakci tajné zprávy ze steganogramu), tajnou zprávu lze reprezentovat změnami fáze mezi jednotlivými fragmenty. Ty však nemohou následovat přímo po sobě, protože tím by vznikly nespojitosti, které by zvukový signál ovlivnily způsobem, který by již lidský sluch zaznamenal (na relativní změny fáze je sluch citlivý). Proto je třeba mezi fragmenty vložit krátké mezery a zajistit spojitost mezi sousedními fragmenty. Tato steganografická technika podle [5] umožní ukrýt 32 bitů na jednu sekundu zvukového záznamu[2].

Rozprostření spektra. [5] Při ukrývání tajných dat ve zvukovém signálu (nosném, původním) je žádoucí, aby signál reprezentující tajná data (de facto vložený signál) měl co možná nejmenší výkonovou hustotu, čehož lze dosáhnout rozprostřením původního spektra vloženého signálu (tedy původní signál zaujímající jen určité oblasti spektra, je rozprostřen po celé dostupné šíři spektra). Technika je založena na násobení vkládaného signálu pseudonáhodným signálem s podstatně vyšší bitovou rychlostí logickou operací *XOR* (exkluzivní disjunkce), daná posloupnost bitů sloužící k rozprostření spektra signálu se nazývá *rozprostírací sekvence*. Výhodou je rovněž fakt, že pro extrakci tajné zprávy je nutná znalost způsobu vytvoření rozprostírací sekvence.

Ukrytí informace v oblasti vyšších kmitočtů má menší vliv na původní (nosný) signál, oproti tomu ukrytí informace v oblasti nízkých kmitočtů zajišťuje vyšší robustnost (odolnost proti odstranění ukryté informace při manipulaci se zvukovým signálem). Rozprostření signálu reprezentujícího tajnou informaci tudíž představuje určitý kompromis mezi těmito vzájemně výlučnými požadavky.[5]

Často využívanou technikou je tzv. **technika *LSB*** (Least Significant Bits, nejméně významné bity), která je do větší hloubky popsána dále.

3.2 Technika LSB

Tato steganografická technika vychází z rozdělení nosiče na dílčí prvky nesoucí vlastní data (ve zvukových souborech se jedná o vzorky), přičemž každý z nich je reprezentován číselnou hodnotou, v tomto kontextu vyjádřenou binárně. U obrazových souborů se může jednat o číselnou reprezentaci odstínu určitého bodu (pokud je obraz ukládán v odstínech šedi), o číselnou reprezentaci barvy určitého bodu (typicky tvořena trojicí hodnot, jednou hodnotou pro každou barevnou složku), u zvu-

[2]Není uvedeno, zda je této informační hustoty dosaženo na jeden kanál nebo na jiný počet kanálů, rovněž se neuvádí, zda je tato hodnota konečná nebo zda může být i vyšší.

kových záznamů o diskrétní vyjádření hodnoty reprezentující změnu hladiny akustického tlaku v určitém čase.

Formálně lze (v některých případech s určitou mírou abstrakce, např. při omezení na jeden kanál) říci, že vlastní data D obsažená v nosiči sestávají z n po sobě následujících vzorků v_n:

$$D = (v_0, v_1, \ldots, v_{n-1}) \tag{3.1}$$

Každý vzorek v je možné zapsat jako binární číslo v součtovém tvaru (a_n představuje koeficient daného řádu, může nabývat hodnoty 0 nebo 1)

$$v = a_{n-1} \cdot 2^{n-1} + a_{n-2} \cdot 2^{n-2} + \cdots + a_0 \cdot 2^0 \tag{3.2}$$

nebo ve zhuštěné formě (s bity vyšších řádů vlevo) jako

$$v = a_{n-1}a_{n-2}\ldots a_0. \tag{3.3}$$

Při použití techniky LSB dochází ve vzorku k nahrazení n bitů nejnižšího řádu (LSB) n bity ukrývané zprávy. Tato úprava je destruktivní, tedy mění původní hodnotu vzorků. Je zřejmé, že míra ovlivnění podoby původního signálu tvořeného posloupností vzorků závisí především na počtu původních LSB, které jsou nahrazeny bity ukrývané zprávy. Z tohoto hlediska je vhodné buď zvolit co nejmenší počet nahrazovaných LSB, nebo – z opačné strany – zvolit nosič s co největší steganografickou kapacitou (tj. záznam s vyšším počtem bitů na vzorek, s vyšším vzorkovacím kmitočtem nebo vyšším počtem kanálů). Využití steganografické kapacity může být dále, v závislosti na povaze ukrývané zprávy, vylepšeno kompresí vkládané zprávy.

Úpravou nejnižších bitů je do původního signálu vnášen šum, což by teoreticky nemuselo vadit z hlediska potenciálního prozrazení při poslechu (třetí strana – útočník – si nemusí rozdílu oproti nosiči v originální formě povšimnout), avšak problémy mohou nastat při hlubší analýze daného steganogramu. Problematická je vyšší úroveň šumu oproti jiným záznamům podobného charakteru, což může být částečně vyřešeno tím, že se vkládaná zpráva před vložením do nosiče zašifruje silnou šifrou, takže zpráva bude mít podobu pseudonáhodných dat, a při vložení do zvukového signálu bude mít charakter blížící se *bílému šumu* (bílý šum je označení pro náhodný signál, který má rovnoměrně rozloženou výkonovou spektrální hustotu [6]). I přesto může steganogram podlehnout stegoanalýze, protože šum vnesený změnou nižších bitů nebude korelovat se šumem obsaženým ve vyšších bitech vzorku[4].

Jedním z možných vylepšení techniky LSB je volba míst v signálu (v časové oblasti), ve kterých budou nejnižší bity nahrazeny. Tím se lze vyhnout významnějšímu ovlivnění částí zvukového záznamu, ve kterých by bylo ovlivnění nápadné. Podobně je možné volit počet nahrazovaných bitů dynamicky v závislosti na průběhu signálu v čase (vzorky s vyšší numerickou hodnotou budou úpravou LSB méně ovlivněny než vzorky s hodnotou blížící se minimu).

Vzorky, ve kterých budou LSB nahrazeny, lze v nejjednodušším případě volit sekvenčně (vzorky jsou upravovány po řadě tak, jak následují ve zvukovém záznamu za sebou). Tento přístup lze částečně z hlediska obtížnější extrakce utajené zprávy útočníkem vylepšit tím, že vzorky, u kterých budou upraveny LSB, se vyberou rovnoměrně z celého rozsahu obsažených vzorků.

Podstatnějšího vylepšení (tj. ztížení extrakce tajné zprávy útočníkem) je možné dosáhnout pseudonáhodnou volbou vzorků, které budou upraveny – volba konkrétních vzorků závisí na zadané vstupní hodnotě (v podstatě se jedná o klíč). Z hlediska utajení je dále vhodné, a to nezávisle na zvolené steganografické technice, vkládanou tajnou zprávu šifrovat. Pokud je navíc požadována jistota[3], že steganogram (resp. utajená zpráva) nebyl při přenosu poškozen nebo úmyslně modifikován, je možné před vložením tajnou zprávu opatřit digitálním podpisem nebo MAC otiskem.

3.3 Steganografická kapacita

Pojem *steganografická kapacita* byl již částečně přiblížen – jedná se o množství informace (nebo správněji o množství dat), které lze do nosiče skrýt, přičemž běžně je tato definice rozšířena o požadavek, že ukrytí daného množství dat do nosiče nebude mít s největší pravděpodobností za následek prozrazení jejich existence.

Při skrývání dat do zvukových záznamů technikou LSB lze steganografickou kapacitu C nosiče definovat jako

$$C = n_b \cdot n_k \cdot f_{vz} \cdot t \text{ [bit]}, \tag{3.4}$$

kde

- n_b je počet bitů na vzorek, které budou nahrazeny bity vkládané tajné zprávy,
- n_k je počet zvukových kanálů v nosiči (zvukovém souboru),
- f_{vz} je vzorkovací kmitočet v Hz (nebo též počet vzorků zaznamenaných, resp. přehrávaných za jednu sekundu),
- t je doba trvání zvukového záznamu v sekundách.

Steganografická kapacita nosiče je neměnná, avšak lze ji za určitých podmínek lépe využít kompresí skrývaných dat (tj. vložit větší množství dat než bez využití komprese). Potenciální zvýšení množství vložitelných dat závisí jednak na zvoleném kompresním algoritmu a jednak na druhu vkládaných dat (například pokud bychom chtěli vkládat textová data, může se množství vložitelných dat zvýšit několikanásobně, oproti tomu při vkládání již komprimovaných dat, např. obrázku ve formátu JPG, by bylo možné vložit pouze nepatrně vyšší množství takových dat).

[3]Přesněji spíše pravděpodobnost, která se limitně bude blížit 1, a to na základě znalosti použitých kryptografických metod a časové/výpočetní složitosti jejich překonání.

3.4 Vhodné nosiče pro techniku LSB

3.4.1 Charakter záznamu

V kap. 3.2 bylo nastíněno, že je vhodné, aby steganografická kapacita nosiče byla co možná nejvyšší (a tudíž aby bylo ovlivnění nosiče vloženou zprávou co nejmenší). Pro příklad – evidentně je vhodnější zvukový stereo (dvoukanálový) záznam s vzorkovacím kmitočtem 48 kHz než jednokanálový záznam s vzorkovacím kmitočtem 22 kHz. Dalším faktorem je i typ a kvalita zvukového záznamu. Z hlediska utajení vložené zprávy je vhodné, aby nosič obsahoval zvukový záznam s co nejmenším množstvím tichých pasáží (vhodná je například dynamická, živá hudba). Stejně tak je z hlediska zanesení šumu změnou LSB výhodné použít záznam, který již obsahuje určité množství šumu (tedy méně kvalitní nahrávku, například hlasový záznam z méně kvalitního záznamového zařízení nebo z rušného prostředí). Ideální by bylo použití nosiče, který by obsahoval pouze bílý šum, nicméně v praxi by toto řešení velmi pravděpodobně vzbuzovalo podezření samo o sobě.

3.4.2 Formáty zvukových souborů

Důležitým hlediskem je rovněž volba konkrétního zvukového formátu, ve kterém bude nosič uložen. Existuje celá škála formátů souborů určených pro ukládání hlasu či hudby (zvukových záznamů obecně, přestože některé formáty mohou být vhodnější pro určitý druh zvukového záznamu), patrně nejrozšířenějším z nich je formát *MP3* využívající ztrátovou kompresi (standard MPEG z roku 1993). Jeho popularita je dána tím, že je široce podporován napříč spektrem programů pro přehrávání a práci se zvukovými záznamy a dále různými fyzickými zařízeními přímo či okrajově určenými pro přehrávání hudby (osobní přehrávače, autorádia, multimediální stolní přehrávače, mobilní telefony aj.). Jeho další podstatnou výhodou oproti nekomprimovaným formátům je významné snížení velikosti výsledných souborů se zvukovým záznamem, což má za následek rychlejší přenos souborů MP3 zejména po síti (především v kontextu Internetu) a nižší nároky na kapacitu úložných médií.

Popularita formátu je rovněž spojena s široce dostupným hudebním obsahem na Internetu, ať už se jedná o prodejní distribuční kanály či nelegálně[4] uložené soubory volně dostupné zdarma ke stažení. Nevýhodou formátu MP3 je zatížení softwarovými patenty (platí jen v některých zemích) a dále nižší kvalita záznamu

[4]Zda je ukládání či stahování hudebních souborů chráněných autorskými právy legální či nikoliv, se liší podle legislativních rámců jednotlivých zemí. V některých zemích není například legální takové soubory bez svolení vlastníka majetkových práv nahrávat na veřejná úložiště nebo je poskytovat třetím osobám, avšak je legální takové soubory stahovat a využívat pro osobní potřebu. Toto ustanovení se týká i České republiky.

při nízkých bitových rychlostech (to samozřejmě závisí i na citlivosti sluchu konkrétního posluchače a kvalitě technického vybavení pro přehrávání záznamu). Rovněž je možné při stejné bitové rychlosti dosáhnout i kvalitnějšího záznamu zvuku v jiných formátech využívajících ztrátovou kompresi (například otevřený formát Ogg Vorbis).

Ztrátová komprese MP3 je postavena na psychoakustickém modelu, který popisuje vnímání zvuku lidským sluchem a lze z něj vyvodit i určité (dříve popsané) nedokonalosti sluchu, a ty využít k odstranění částí ve zvukovém záznamu, které nepovedou k podstatným změnám v kvalitě záznamu, ale sníží celkovou velikost výsledného souboru. Z podstaty ztrátové komprese však plyne, že část původně obsažené informace je odstraněna, a tudíž není možné zachovat plnou kvalitu původního záznamu. Z tohoto důvodu se používají zejména v profesionální sféře či ve sféře náročných posluchačů hudby další formáty, které nejsou postiženy nevýhodami ztrátové komprese.[5]. Rozšířené jsou v této oblasti mj. formáty *WAV* (podrobně bude popsán dále) a *FLAC* (*Free Lossless Audio Codec*, svobodný bezeztrátový audiokodek). Formát (kodek) FLAC sice neposkytuje tak vysokou úroveň komprese jako kodeky využívající ztrátovou kompresi, avšak zachovává plnou kvalitu původního zvukového signálu.

3.4.3 Formát WAV

Formát WAV, definovaný společnostmi Microsoft a IBM a používaný jako jeden z hlavních formátů pod OS rodiny Windows[7], představuje *kontejner*, tedy definuje určitý rámec stanovující, jakým způsobem budou v rámci souboru ukládána vlastní zvuková a případně i jiná data. Může obsahovat komprimovaná i nekomprimovaná data (za použití různých kodeků) a rovněž i metadata, zvukový záznam je však většinou ukládán pouze nekomprimovaně, a to ve formátu (L)PCM[6] (Linear Pulse Code Modulation, lineární impulsní kódová modulace). Může obsahovat i vícekanálový záznam včetně popisu umístění jednotlivých zdrojů zvuku (v případě rozšířeného formátu WAV)[7], nicméně v praxi se lze převážně setkat se soubory obsahujícími jednokanálovou či dvoukanálovou zvukovou stopu. Z tohoto důvodu (a s ohledem na cíl této práce, tj. realizace laboratorní úlohy, bude popis formátu WAV dále omezen na tuto skutečnost. Ze stejných důvodů budeme dále předpokládat přítomnost pouze jedné zvukové stopy uvnitř souboru, přestože obecně je možné v rámci jednoho souboru uložit i více samostatných stop.

[5]Mají ale jiné nevýhody, především že záznam zabírá více místa než u ztrátových kompresí, nicméně vždy je nutné, jako u většiny technických oblastí, zvážit požadované vlastnosti a oblast nasazení a podle toho správný formát vybrat

[6]LPCM je konkrétním druhem obecnější PCM, většinou se ale pod pojmem PCM myslí právě LPCM[8]

Formát WAV představuje konkrétní aplikaci obecnějšího formátu *RIFF* (Resource Interchange File Format, souborový formát pro výměnu zdrojů). Formát RIFF je založen na uspořádání dat a metadat do *bloků* (*chunks*), které jsou dále členěny do *podbloků* (*subchunks*). Začátek každého bloku nebo podbloku je identifikován čtveřicí znaků (tento identifikátor je označován jako *FOURCC*) a označuje typ dat, která jsou v daném bloku uložena, přičemž identifikátory sestávající ze čtyř velkých písmen jsou považovány za vyhrazené. Data uvnitř každého bloku či podbloku musí být při zpracování interpretována podle identifikátoru daného bloku (tj. podle konkrétního významu dat v bloku). Formát bloku nebo podbloku je následující[10]:

- čtyřznakový identifikátor typu bloku – 4 B,
- délka datové posloupnosti obsažené v bloku uvedená v bajtech (číselná hodnota je 32bitové celé číslo little endian bez znaménka; hodnota nezahrnuje čtyřznakový identifikátor a čtyři bajty pro označení délky bloku) – 4 B,
- následují vlastní zvuková data (vzorky) s proměnnou délkou,
- výplňový bajt pro zarovnání na sudou délku (doplní se pouze v případě, že je délka bloku lichá)

Formát RIFF definuje mj. identifikátory bloků `INFO`, `JUNK` a `PAD` [7]:

- Identifikátor `INFO` označuje blok metadat sloužících k označení autora dat, názvu díla apod.
- Identifikátor `JUNK` slouží k označení dat, která nemají být použita. Přepsáním identifikátoru určitého bloku/podbloku na `JUNK` lze tedy de facto odstranit data obsažená v tomto bloku. Nejedná se o odstranění ve smyslu smazání dané sekvence bajtů ze souboru, ale spíše o jejich zneviditelnění v rámci programu, který s RIFF souborem pracuje. Dalším možným využitím tohoto identifikátoru je vytvoření místa uvnitř bloku pro budoucí dodatečné zapsání nových dat, aniž by při zápisu nových dat bylo nutné přepisovat celý soubor.
- Identifikátor `PAD` má podobný význam jako `JUNK` (vytvoření nevyužitého místa uvnitř souboru).

Význam těchto typů bloků (zejména `JUNK` nebo `PAD`) naznačuje, že by bylo možné do souboru WAV poměrně jednoduchým způsobem ukládat tajná data vložením takovýchto bloků (a změnou hodnoty délky patřičného nadřazeného bloku). To by však poskytovalo utajení dat pouze na velmi triviální úrovni – utajení by bylo účinné prakticky pouze vůči technicky neznalým osobám. Pokud by ovšem k souboru získala přístup osoba se znalostí dané problematiky, utajení by bylo velmi pravděpodobně prolomeno (tím spíše, pokud by utajovaná data byla pouze otevřeným textem, protože by jej bylo možné zobrazit pouhým otevřením souboru v prohlížeči).

Problém by rovněž představovala strojová kontrola (analýza) souboru. Důvěrnost dat by se sice dala velmi snadno vyřešit za pomoci šifrování, nicméně takto vložené bloky dat by nejspíše – zejména při strojové kontrole – budily podezření. Z těchto důvodů se nejedná o příliš vhodnou metodu pro ukládání tajných dat.

Důležité je poznamenat, že podposloupnosti bajtů v rámci jednotlivých bloků či podbloků se mohou odlišovat *endianitou* (uspořádáním bajtů ve slově tvořeném posloupností bajtů od nejvíce po nejméně významné), takže je nutné je interpretovat odpovídajícím způsobem. Např. identifikátory typu bloků (FOURCC) mají endianitu typu big endian (nejvýznamnější bajt na první pozici), zatímco délky datových posloupností jsou typu little endian (nejméně významný bajt na první pozici); implicitní endianita ve WAV souboru je little endian [9]. V následujícím popisu struktury souboru WAV bude endianita konkrétních částí dat v případě potřeby zkráceně značena jako *LE* (little endian) nebo *BE* (big endian).

Vnitřní struktura formátu WAV

Blok `RIFF`. V souladu se strukturou typu RIFF je v souboru WAV na počátku umístěn blok `RIFF`, který představuje z hlediska struktury souboru logicky nejvyšší blok. Záhlaví bloku je tvořeno těmito položkami:

- identifikátor s ASCII hodnotou `RIFF` (4 B, BE),
- délka bloku v bajtech (4 B, LE),
- formát souboru – ASCII hodnota `WAVE` (4 B, BE)

Podblok `WAVE` obsahuje podbloky „`fmt` “ (mezera za znaky `fmt` je skutečně obsažena) a `data`. Struktura a význam jednotlivých polí (podposloupností bajtů) bloku `fmt` je následující:

- identifikátor podbloku – ASCII řetězec `fmt` (4 B, BE),
- velikost podbloku `fmt` (4 B, LE),
- typ vzorků (hodnota `1` reprezentuje lineární kvantování – LPCM, jiné hodnoty než 1 znamenají, že je použita komprese),
- počet kanálů n_k,
- vzorkovací kmitočet (počet vzorků přehrávaných za 1 sekundu) f_{vz} (v Hz),
- počet přehraných bajtů za sekundu (vypočte se jako $n_k \cdot n_b \cdot f_{vz}/8$),
- velikost vzorku[7] v bajtech (vypočte se jako $n_b \cdot n_k/8$),
- počet bitů na vzorek n_b.

[7] V tomto případě se nemusí jednat přímo o jeden vzorek coby číselnou hodnotu v daném čase, ale v případě vícekanálové zvukové stopy je v tomto kontextu pod pojmem *vzorek* chápáno více souvisejících hodnot vzorků, které jsou přehrávány ve stejném čase, avšak každý pro patřičný kanál.

Podblok `data` je členěn tímto způsobem:

- identifikátor podbloku – ASCII řetězec `data` (4 B, BE),
- velikost bloku v bajtech l_B (vypočte se jako $n_k \cdot n_b \cdot n_{vz}/8$, kde n_{vz} je počet všech vzorků pro jeden kanál),
- vlastní data (posloupnost vzorků) v celkové délce specifikované hodnotou l_B. V případě, že se jedná o 8bitové vzorky, mají bajty reprezentující hodnoty jednotlivých vzorků význam celých nezáporných 8bitových čísel (v rozsahu $[0; 255]$); pokud se jedná o 16bitové vzorky, mají dvojice bajtů reprezentující hodnoty jednotlivých vzorků význam celých 16bitových čísel se znaménkem ve dvojkovém doplňkovém kódu (tj. číslo v rozsahu $[-32768; 32767]$).

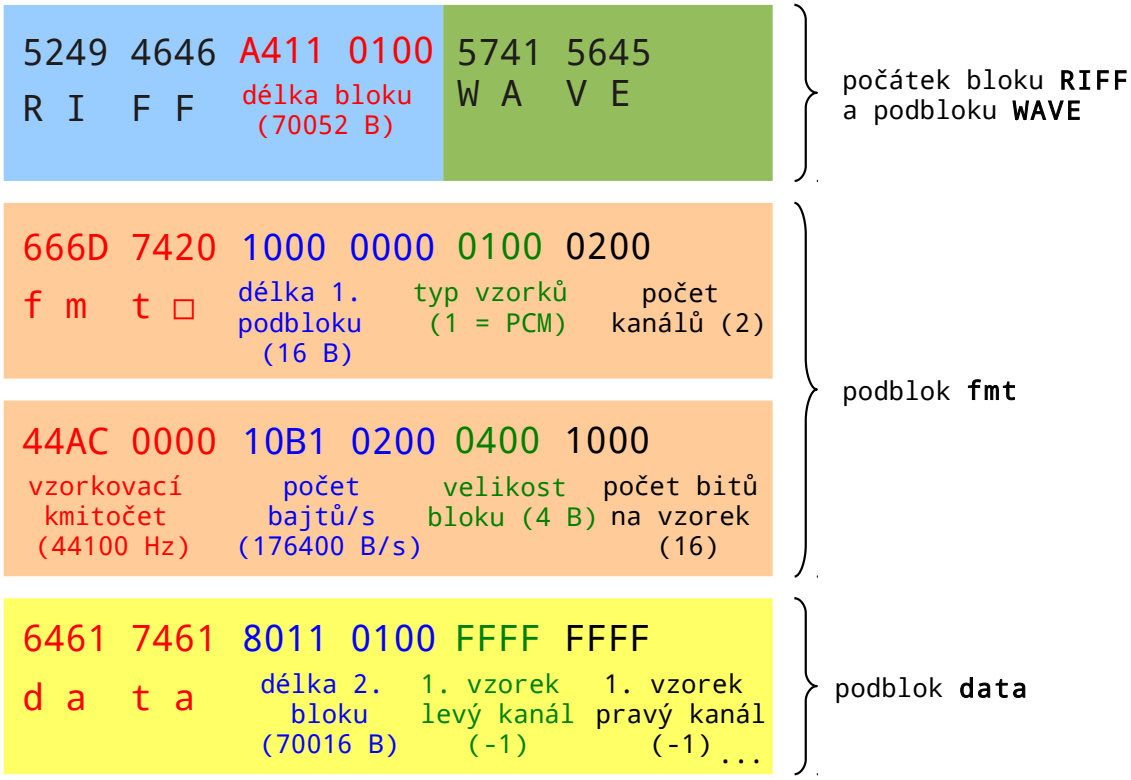
Za těmito podbloky mohou případně následovat další podbloky (každý začíná čtyřbajtovým 4B identifikátorem bloku, dále obsahuje hodnotu délky následujících dat a vlastní data).

Příklad. Pro lepší přiblížení struktury souboru WAV je znázorněna a popsána struktura souboru WAV na konkrétním příkladu – souboru `ding.wav`, který je součástí instalace systému Microsoft Windows 7. Obrázek 3.1 znázorňuje obsah prvních 48 bajtů tohoto souboru, tj. část od identifikátoru `RIFF` až po první vzorek pro pravý a levý kanál (další, již nezobrazené bajty, představují následující PCM vzorky). Význam konkrétních skupin bajtů je popsán na obr. 3.2.

Z důvodu formátování jsou bajty na obrázcích uváděny po dvojicích a pro přehlednost jsou jednotlivé skupiny bajtů a bloky/podbloky barevně odlišeny (barevné značení na obou obrázcích odpovídá stejným skupinám bajtů).

5249	4646	A411	0100	5741	5645	666D	7420
1000	0000	0100	0200	44AC	0000	10B1	0200
0400	1000	6461	7461	8011	0100	FFFF	FFFF
...							

Obr. 3.1: Příklad obsahu souboru WAV



Obr. 3.2: Rozbor obsahu souboru WAV

4 LABORATORNÍ ÚLOHA

Jedním z hlavních cílů této práce je vypracování zadání laboratorní úlohy zaměřené na skrývání tajných dat do zvukových záznamů a vytvoření programu, který skrývání dat dle zadaných požadavků umožní.

4.1 Požadavky

Laboratorní úloha, a pro její účely vytvořený program, vychází z těchto skutečností:

- Úloha bude součástí počítačových cvičení v rámci předmětu *Bezpečnost informačních systémů*.
- Cílem úlohy je prozkoumat vliv ukryté informace na kvalitu zvukového záznamu (především do jaké míry je záznam ovlivněn množstvím ukrývané informace a jak se projeví skrytí dat v různých typech zvukových záznamů a různá nastavení způsobu ukrytí dat).
- Úloha bude probíhat na počítačích s OS Microsoft Windows 7.
- Čas na zpracování úlohy je stanoven na přibližně 90 minut.
- Od studentů nebudou požadovány žádné výstupy (protokoly).

4.2 Cíle úlohy

V rámci úlohy by si studenti měli ověřit, jaký vliv má hustota ukrývané informace na výsledný zvukový záznam, tedy ovlivnění jeho kvality a z toho plynoucí bezpečnosti ukrývaných dat (z hlediska zkoumání zvukového záznamu třetí stranou – posluchačem). Dále by si měli ověřit vliv kryptografických ochran (šifrování) na bezpečnost ukrývaných dat.

4.3 Didaktický pohled

Při návrhu úlohy je třeba splnit jak obsahové požadavky (vlastní náplň úlohy), tak i požadavky na přínos pro studenty, kterým bude úloha zadána. Studenti mají s danou problematikou pravděpodobně pouze teoretickou zkušenost z výuky, případně ilustrovanou na příkladech, avšak lze předpokládat, že až při vlastní praktické zkušenosti si více uvědomí jednotlivé aspekty či souvislosti (což by mělo být cílem právě laboratorních úloh či počítačových cvičení).

Zadání úlohy by mělo být vyvážené, aby korelovalo s tím, co studenti již z výuky znají – tedy aby zbytečně nesuplovalo úlohu části přednášek, avšak zároveň studentům oblast steganografie znovu alespoň ve stručnosti přiblížilo, včetně klíčových

faktů a mechanismů. Dále by mělo zadání v úvodu do problematiky zmínit specifika skrývání dat ve zvukových záznamech a popsat způsoby, kterými bude ukrývání dat provádět program použitý v rámci cvičení. Již zmíněná vyváženost by se měla týkat i vlastního programu, zejména z hlediska návrhu grafického uživatelského rozhraní (GUI) – blíže rozvedeno v kap. 5.2 – tj. především volbě a rozmístění ovládacích prvků a případných dialogů nebo podružných oken.

V rámci práce s programem by si studenti měli vyzkoušet jeho použití pro různé vstupy a způsoby nastavení – zvukové záznamy různého typu (hudba, mluvené slovo, nahrávka se šumem apod.), více druhů skrývaných dat, různé způsoby rozprostření ukrývaných dat, a především různá hustota ukrývané informace. Na základě těchto vstupů by si měli uvědomit, jak bude ovlivněn charakter výsledného zvukového záznamu a jaké jsou výhody a nevýhody konkrétního řešení. Rovněž by se měli zamyslet nad otázkou bezpečnosti skrývaných dat a vlivem šifrování.

Data, na kterých si studenti mohou výše uvedené náležitosti vyzkoušet, jsou součástí úlohy. Jedná se o sadu textových souborů a fotografií JPEG o různých velikostech, a dále zvukových souborů WAV různého charakteru (mluvené slovo, hudba, ticho, bílý šum aj.). Soubory, které jsou součástí úlohy, jsou podrobně popsány v příloze B.

Vlastní skrývání dat se provádí za využití techniky LSB ve zvukových souborech WAV, přičemž je možné nastavit množství ovlivněných nejméně významných bitů, způsob rozprostření ukrývaných dat ve zvukovém záznamu, volitelně aplikovat šifrování a případně i využít kompresi dat pro snížení vlivu ukrývaných dat na zvukový záznam (blíže popsáno v kap. 5 a příloze A).

Technika LSB v kombinaci se soubory WAV byla zvolena jako určitý kompromis – technika LSB poměrně přímočaře ilustruje možnosti ukrývání dat a změny ve výsledném souboru v porovnání s originálem, avšak zároveň je možné ji použít ne zcela triviálním způsobem a tajná data ukrýt přinejmenším před lidským útočníkem (tj. při poslechu výsledného souboru, případně základní analýze jeho obsahu bez využití pokročilejších nástrojů).

Na druhou stranu vždy dojde k určité změně (poškození) původního souboru, a tedy možnost odhalení skrytých dat, resp. skutečnosti, že soubor byl nějakým způsobem upraven, závisí na konkrétních okolnostech. Z hlediska vlastního skrytí dat se především jedná o

- **charakter nosiče** – přirozený obsah šumu v záznamu, obsah hlasitých či dynamických, nebo naopak tichých pasáží, steganografická kapacita, počet bitů na vzorek,
- **vlastnosti souboru se skrývanými daty** – jestli se jeho obsah blíží spíše pseudonáhodnému charakteru, nebo je do určité míry pravidelný, dále záleží na velikosti souboru a případně obsaženém množství redundantní informace,

- **aplikaci komprese nebo šifrování** – šifrování může změnit charakter poškození záznamu (vnesené změny se podobají bílému šumu), podobně se může projevit komprese, a zejména může u vhodného typu dat (např. text) razantně snížit míru poškození záznamu, protože se do něj ve výsledku vkládá značně snížené množství dat,
- **hloubku modifikace** (počet změněných LSB ve vzorku) – zbytečně velká hloubka modifikace může zvýšit riziko odhalení manipulace se záznamem,
- **způsob rozprostření** – v závislosti na povaze nosiče a skrývaných dat se bude ovlivnění výsledného záznamu lišit (mimo dalších faktorů) na základě způsobu rozprostření.

Z hlediska poslechové analýzy výsledného záznamu možnost odhalení změn (postřehnutí poškození záznamu) závisí na dalších faktorech, zejména na

- **kvalitě přehrávacího zařízení** (zvukové karty, propojovacích kabelů, sluchátek či reproduktorů), příp. přítomnosti rušivých vlivů,
- **zesílení** přehrávaného zvukového záznamu, respektive hlasitosti na výstupu zvukové aparatury,
- **přítomnosti vlivů, které znesnadňují soustředění posluchače** (především hladina okolního hluku),
- **vlivech působících na psychické rozpoložení posluchače** (stres, únava) a jeho fyzické dispozice (citlivost sluchu).

Všechny uvedené náležitosti by studenti rovněž při řešení úlohy měli brát v úvahu. Vzhledem k tomu, že po vypracování úlohy nejsou požadovány žádné výstupy (výpočty, protokoly apod.), byla praktická část úlohy koncipována formou pokynů, které jsou následně doplněny otázkami nebo nápovědou, na co by se studenti měli blíže zaměřit a čeho by si měli všímat. Při zpracování úlohy by se měli snažit na tyto otázky odpovědět, vedoucí cvičení si může např. namátkou odpovědi kontrolovat a tak se ujistit, zda studenti problematice porozuměli.

K daným otázkám jsou vypracovány odpovědi, resp. poznámky k daným otázkám. Tyto poznámky jsou určeny vedoucím cvičení, měly by mj. přiblížit charakter očekávaných odpovědí a případně pomoci s nasměrováním studenta k pochopení jevu, na který se zaměřuje daná otázka.

4.4 Struktura úlohy

Zadání úlohy má následující strukturu:

1. teoretický úvod do problematiky steganografie
   a) specifika ukrývání dat ve zvukových záznamech
   b) princip techniky LSB

   c) základní popis formátu WAV
2. popis programu a jeho funkcí
3. ukrývání dat do zvukových souborů pro různé další vstupy a nastavení

Teoretický úvod do problematiky steganografie je omezen pouze na zásadní body (studenti již pojem znají z přednášek). Úvod dále popisuje, jaké jsou možnosti skrývání dat do zvukových záznamů a uvádí rysy techniky LSB. V teoretické části je popsána i základní struktura souborů WAV a způsob uložení číselných hodnot vzorků do bajtů v datové části souboru WAV.

Po teoretickém úvodu následuje popis programu Stego, který je v úloze pro skrývání dat využíván. Seznamuje studenty s vlastnostmi programu a s jeho ovládáním, pokud by studenti potřebovali bližší informace, jsou odkázáni na nápovědu (uživatelskou příručku), která je součástí programu – jedná se o detailní rozšíření základních informací uvedených v zadání úlohy (její znění je uvedeno v příloze A).

Následuje vlastní zadání úlohy, které se zaměřuje na skrývání dat do zvukových záznamů. Nejdříve se studenti prakticky seznámí s programem a jednotlivými způsoby rozprostření dat do záznamu obsahujícího ticho (takže je v grafické reprezentaci ihned vidět, jak se změny projeví, pochopitelně si studenti změny ověří i poslechem). Skrývání dat si vyzkouší i do záznamu obsahujícího bílý šum a mohou vyhodnotit, jak se ovlivnění záznamu různými způsoby ukrývání liší, zda se poškození svým charakterem podobá bílému šumu, vnesení nového tónu nebo jinak. Rovněž si ověří, jak se projeví šifrování dat z hlediska bezpečnosti dat (obsahu výsledného záznamu) i poškození záznamu z poslechového hlediska.

V rámci úlohy si rovněž studenti ověří, jak moc jsou sluchem postřehnutelné změny pro různé hloubky modifikace, a pokusí se najít hloubku modifikace, která ještě není sluchem snadno objevitelná (zde je nutné vzít v úvahu dříve zmíněné faktory, které mají vliv na poškození záznamu, a zejména na faktory ovlivňující jeho analýzu poslechem, tj. nelze stanovit absolutní hloubku modifikace, při které bude obecně poškození nezjistitelné). Dalším z úkolů je porovnat, jak se ovlivnění záznamů liší pro soubory WAV s 8bitovými a 16bitovými vzorky a pro další typy záznamu, do kterých zatím nebylo skrytí dat provedeno (především soubor s hudbou, mluveným slovem a nahrávkou z mobilního telefonu obsahující šum). V závěru zadání si studenti mohou vyzkoušet další kombinace vstupních dat, různých nosičů WAV a různých nastavení dle vlastního uvážení, a mohou s nimi experimentovat.

5 PROGRAM PRO LABORATORNÍ ÚLOHU

V kap. 4 byly stanoveny požadavky a cíle laboratorní úlohy, jejíž klíčovou součástí je program realizující skrývání dat libovolného charakteru (z uživatelsky vybraného souboru) do zvukového souboru typu WAV. V této kapitole jsou popsány vlastnosti a způsob implementace tohoto programu.

5.1 Funkční požadavky

Z hlediska dekompozice na jednotlivé funkční oblasti byl program navržen tak, aby umožnil provádět především dále popsanou množinu operací:

5.1.1 Skrývání dat

Nastavení počtu ovlivněných LSB ve vzorku. Hloubku modifikace lze ručně nastavit podle požadavku uživatele, spolu s **nastavením způsobu rozprostření** ukrývaných dat ve zvukovém záznamu. Při načítání vstupních dat je prováděna kontrola na steganografickou kapacitu nosiče (viz část 5.4.3) a na základě jejího výsledku je nastavena minimální hloubka skrývání (počet přepisovaných LSB ve vzorku). Tato hodnota je nastavena jako výchozí, takže jsou využity dostupné vzorky v záznamu s co nejnižší možnou měrou ovlivnění.

Zobrazení informací o zvukovém souboru a výpočet steganografické kapacity. Program uživateli zobrazí informace o zvukovém záznamu – délku záznamu, vzorkovací kmitočet, počet bitů na vzorek, počet kanálů a teoretickou steganografickou kapacitu (může být ve skutečnosti i vyšší při použití komprese; jedná se ale o teoretickou hodnotu, protože při využití celé steganografické kapacity by došlo k totálnímu přepsání původních zvukových dat ve vzorcích). Steganografická kapacita vybraného souboru je vypočtena na základě počtu kanálů, vzorkovacího kmitočtu, počtu bitů na vzorek a počtu vzorků v nosiči.

Volitelné šifrování skrývaných dat. Šifrování dat se provede za podmínky, že uživatel aktivuje volbu šifrování a zadá heslo, které zprostředkovaně slouží jako šifrovací klíč. Klíč je vytvořen na základě hesla za využití hašovacího algoritmu SHA2-256 (*Secure Hash Algorithm*), takže heslo ve formě textového řetězce může mít prakticky libovolnou délku a vždy je transformováno na klíč stejné délky. Vlastní šifrování se provádí symetrickou blokovou šifrou AES-256 (*Advanced Encryption Standard*). Z hlediska zamýšleného použití (laboratorní úloha) by bylo možné zvolit

i slabší šifru, nicméně z hlediska implementace nepřináší použití šifry AES žádné komplikace oproti použití jiné (slabší) šifry.

Volitelná komprese skrývaných dat. Komprese dovoluje – v závislosti na typu ukrývaných dat – lepší využití steganografické kapacity, tj. ukrytí vyššího množství dat při zachování stejné hloubky modifikace, anebo případně zmenšení potřebné hloubky modifikace při ukrytí stejného množství dat. Využita je komprese Bzip2.

Volba rozprostření dat ve zvukovém záznamu. Rozprostřením dat ve zvukovém záznamu je myšlen způsob výběru konkrétních vzorků, jejichž LSB budou změněny. První (výchozí) volbou je nejpřímočařejší způsob – sekvenční výběr vzorků, tj. u potřebného počtu po sobě následujících vzorků jsou dle zadané hloubky modifikace přepsány jejich LSB. Další volbu představuje rovnoměrné rozložení – vzorky určené k modifikaci jsou vybrány rovnoměrně od začátku do konce záznamu a mezi nimi jsou případně ponechány stejně velké skupiny neovlivněných vzorků. Poslední možností je pseudonáhodné rozprostření. V tomto případě jsou pozice ovlivněných vzorků stanoveny na základě zadaného hesla. Vlastností tohoto způsobu rozprostření je, že budou ovlivněny pouze určité vzorky ze záznamu, a to ne nutně v sekvenčním pořadí. Bez znalosti hesla může tedy útočník teoreticky určit, které vzorky byly ovlivněny, avšak již neurčí, v jakém pořadí je třeba spojit n-tice LSB ovlivněných vzorků pro získání utajené zprávy.

Pseudonáhodné rozprostření se svým principem blíží aplikaci šifry založené na permutaci n-tic vstupních bitů dat, ale sama o sobě nemusí poskytovat dostatečnou úroveň důvěrnosti dat. V závislosti na hloubce modifikace a velikosti a typu skrývaných dat může být případný útočník schopen zjistit, jaká data (zpráva) byla uložena, nebo alespoň odhadnout, o jaký druh zprávy se jedná. Příkladem může být situace, kdy ukryjeme textovou zprávu se sedmibitovým nebo osmibitovým kódováním (ASCII, Latin2, CP-1250 apod.) s nastavenou hloubkou modifikace 8 nebo 16 LSB na vzorek.

V takovém případě budou při zobrazení obsahu výsledného souboru WAV vidět bez jakékoliv hlubší analýzy jednotlivé znaky skrývaného textu. Pokud bude zpráva velmi krátká, může být útočník schopen odhadnout její obsah například extrakcí znaků, které spadají do abecedy (písmen, příp. číslic), a může se pokusit poskládat jednotlivé znaky v jiných pořadích a části výsledného řetězce porovnat se slovníkem. Přitom samotné permutace nemusí dělat hrubou silou, ale například podle charakteristiky jazyka, ve kterém je skrytá zpráva napsána (čímž eliminuje nesmyslné kombinace a sníží čas potřebný k prolomení ochrany). Pro delší zprávy by pochopitelně byla nutná sofistikovanější metoda útoku, zejména z důvodu příliš vysoké výpočetní náročnosti zmíněné jednoduché metody.

Přehrávání zvukových záznamů. Je možné přehrát jak originální zvukový záznam, tak i záznam obsahující skrytá data, a subjektivně porovnat, jak moc se vložením skrytých dat změnila kvalita záznamu, a rovněž posoudit charakter změny.

5.1.2 Extrakce skrytých dat

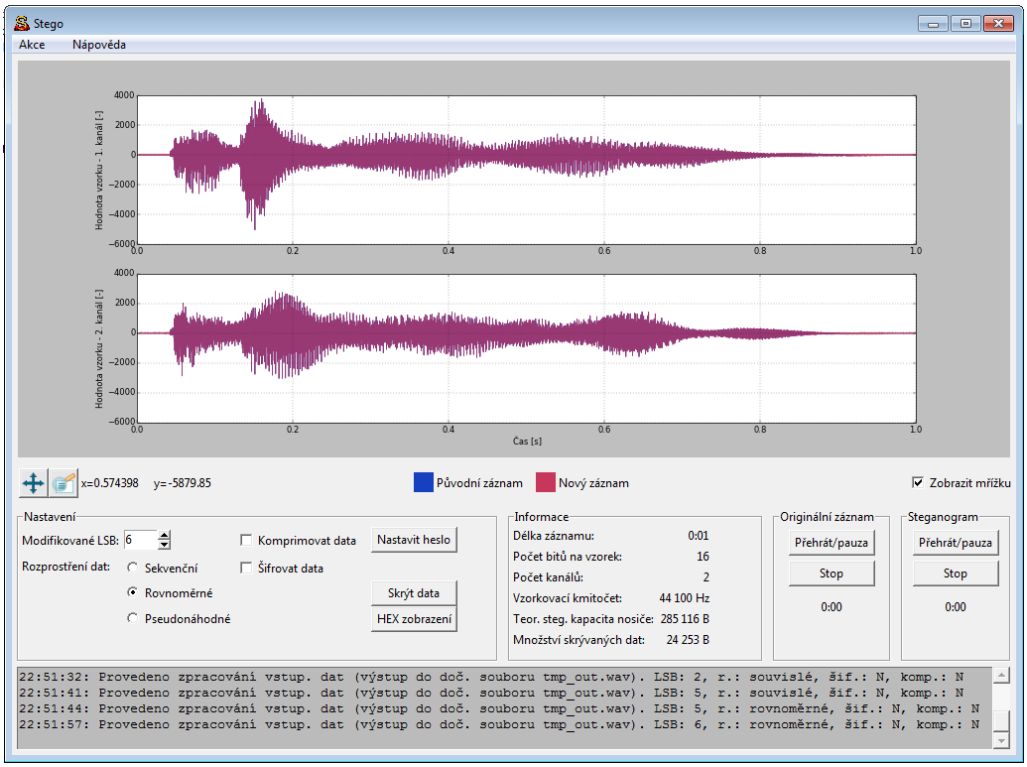
Program musí umožnit i inverzní operaci ke skrytí dat, tedy extrakci skrytých dat z nosiče. Při extrakci může být zapotřebí zadat heslo (pokud jsou skrytá data šifrována), avšak program sám o sobě neurčí, zda jsou data šifrována či nikoliv (šifrování proběhne ve fázi skrývání dat až jako poslední krok přípravy dat předcházející vlastnímu skrytí, a šifrována jsou jak vlastní data, tak i záhlaví, tzn. ani na jeho základě nelze rozhodnout, jestli jsou data šifrována či nikoliv). V případě, že byla skrytá data komprimována, je tato skutečnost detekována automaticky ze záhlaví skrytých dat. Součástí záhlaví skrytých dat je dále i haš (plnící funkci kontrolního součtu), který umožní detekovat, jestli nebyla data poškozena nebo úmyslně změněna, případně jestli nebylo zadáno špatné heslo k dešifrování.

5.1.3 Porovnání vstupního a výstupního záznamu

Tato funkce umožňuje vizuální porovnání průběhů zvukového signálu v čase, a to pro původní záznam a nově vytvořený záznam obsahující skrytá data. Cílem této funkce je poskytnout lepší představu o tom, jaké změny ukrytí dat způsobilo. Mimo grafické reprezentace zvukových signálů nabízí program možnost prohlédnout si obsah původního a nově vytvořeného souboru WAV (kompletní obsah vč. identifikátorů bloků a metadat). Zobrazeny jsou jednak bajty v textové reprezentaci, a dále ve formě šestnáctkových (hexadecimálních) čísel, přičemž odlišné bajty na stejné pozici jsou barevně zvýrazněny.

5.2 Grafické rozhraní

Návrh grafického uživatelského rozhraní (GUI, Graphical User Interface) je jednou z důležitých (avšak nezřídka přehlížených) součástí návrhu programů. Uživatelské rozhraní by mělo co nejlépe odrážet zamýšlené funkce programu a ideálně umožnit uživateli znalému problematiky program použít intuitivně, bez nutnosti studovat návod k použití nebo nápovědu (nicméně nápověda, resp. uživatelská příručka – uvedena v příloze A –, je i přesto nedílnou součástí programu, a to pro případ, že by si uživatel nevěděl rady s určitým postupem nebo by jej zajímaly podrobnější informace o programu, které nemusí být na první pohled zjevné).



Obr. 5.1: Hlavní okno programu Stego

Program je navržen tak, aby většina funkcí byla dostupná přímo z hlavního okna. Ovládací prvky jsou rozčleněny dle významu do několika oblastí (bloků), součástí hlavního okna je i prvek pro grafickou reprezentaci zvukových signálů a textové pole s protokolem zachycujícím, jaké uživatelské činnosti byly vykonány a v jakém čase (takže uživatel má přehled např. o tom, jaké kombinace parametrů a s jakými soubory již vyzkoušel).

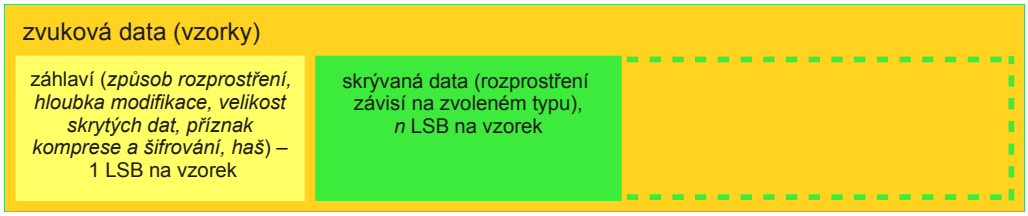
Vzhledem k tomu, že program je navržen pro laboratorní úlohu, je žádoucí, aby bylo možné provádět různé změny nastavení co nejrychleji. Proto jsou v hlavním okně obsaženy i prvky (tlačítka, zaškrtávací pole apod.), které usnadní přímou změnu parametrů, aby tak bylo možné co nejrychleji pozorovat změny, které ve zvukovém záznamu nastaly.

Podružné funkce jsou soustředěny do oddělených oken (dialogů). Jedná se o dialog pro zadání hesla, zobrazení obsahu původního a nově vytvořeného souboru (HEX prohlížeč), zobrazení informací o souboru před extrakcí skrytých dat a dále informační okno zobrazující průběh zpracování dat (skrývání, extrakce).

5.3 Struktura ukládaných dat

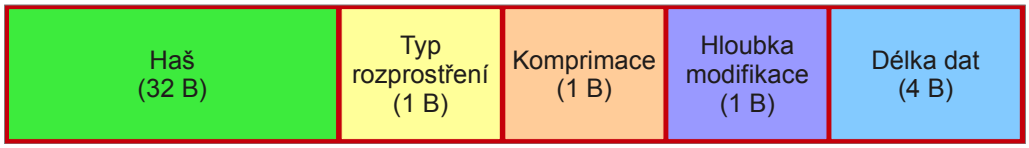
Před vlastními tajnými daty je do vzorků ve vstupním souboru WAV nejprve skryto záhlaví, které nese informace o skrývaných datech (blíže znázorněno na obr. 5.2). Délka záhlaví se liší podle toho, zda je šifrované či nikoliv (šifrováním se délka záhlaví zvýší o inicializační vektor (16 B), informaci o délce šifrovaných dat (4 B) a případné doplnění šifrovaných dat na počet bajtů dělitelný 64). Záhlaví (nešifrované) má tuto podobu (blíže viz obr. 5.3):

- **Haš** (32 B) – s jeho pomocí lze při extrakci skrytých dat určit, zda nebyl soubor poškozen, popř. v kombinaci se šifrováním i zda nebyl úmyslně modifikován, respektive zda bylo při extrakci zadáno správné heslo pro dešifrování. Hašování se provádí hašovací funkcí SHA2-256.
- **Typ rozprostření** (1 B) – sekvenční, rovnoměrné, pseudonáhodné.
- **Příznak komprimace** (1 B) – 0 = bez komprimace, 1 = komprimace použita.
- **Hloubka modifikace** (1 B) – počet nahrazovaných LSB na vzorek.
- **Délka** vlastních skrytých dat následujících bezprostředně za záhlavím (4 B).



Obr. 5.2: Struktura zvukových vzorků

Informace ze záhlaví jsou využity při extrakci dat, aby bylo možné extrakci provést správně, tedy načíst a spojit LSB ze správných vzorků a případně data dekomprimovat. Dalším účelem informací ze záhlaví je možnost ověřit, že data nebyla poškozena, resp. že v případě šifrování bylo zadáno správné heslo.



Obr. 5.3: Záhlaví skrývaných dat

5.4 Implementace

Program Stego je implementován v programovacím jazyce Python[11], přičemž většina funkcí, které program využívá (mj. funkce pro kompresi Bzip2, hašování, sada komponent pro tvorbu GUI Tkinter, funkce pro práci s konfiguračními soubory aj.), je již obsažena ve standardní instalaci. Program je tvořen řadou vzájemně propojených tříd, přičemž z funkčního hlediska se jedná o dvě skupiny tříd: První skupinu tvoří třídy zajišťující vlastní skrývání/extrakci dat do souborů WAV a podpůrné funkce (generování pozic v rámci vzorků při skrývání pseudonáhodnou metodou, správa záhlaví skrývaných dat, šifrování, kontrola vstupních dat). Do druhé skupiny dle tohoto rozdělení spadají třídy realizující GUI – jedná se o implementaci oken/dialogů, některých prvků GUI (prohlížeč dat), funkcí pro správu uživatelských vstupů (načtení vstupních souborů, kontroly vstupů, aktivace/deaktivace ovládacích prvků na základě výsledků kontrol), informačních a zobrazovacích funkcí a funkcí pro přehrávání zvukových záznamů.

Grafické rozhraní využívá sady komponent Tkinter[12][13] (implementace sady komponent Tk pro jazyk Python), resp. částečně Tix (rozšířená sada Tkinter), která je součástí instalace jazyka Python a poskytuje prvky jako například rámce, tlačítka, textová pole, nabídky, textové popisky a přepínače. Pro přehrávání zvukových záznamů ze souborů WAV je využita knihovna PyAudio[14], pro šifrování šifrou AES-256 knihovna PyCrypto. Zobrazování průběhů zvukového signálu zajišťuje knihovna Matplotlib[15].

Dále jsou popsány jednotlivé oblasti implementace a třídy, které jsou v nich využity. Rovněž jsou popsány stěžejní metody v rámci tříd, které souvisí s realizací daných funkcí a hlavní myšlenky, resp. algoritmy. Popis programu z uživatelského hlediska se nachází v příloze A.

Protože Python je interpretovaný jazyk, je za normálních okolností potřeba mít pro spuštění programu nainstalované prostředí jazyka. Aby bylo možné program spustit i bez tohoto prostředí, byl za pomoci nástroje `cx_freeze`[19] vytvořen spustitelný soubor doprovázený sadou potřebných knihoven a dalších souborů.

5.4.1 Záhlaví skrývaných dat

Záhlaví skrývaných dat je spravováno pomocí třídy `Header`. Třída definuje tyto hlavní metody:

- `load` – načtení dat záhlaví ze zhuštěného tvaru (posloupnosti bajtů)
- `update` – nastavení nebo úprava údajů v záhlaví
- `getdata_struct` – vrácení obsahu záhlaví ve formě datové struktury *slovník* (dictionary)

- `getdata` – vrácení posloupnosti bajtů obsahující údaje ze záhlaví
- `getlength` (`getlength_encrypted`) – vrácení délky záhlaví (příp. délky záhlaví, pokud je šifrováno)

V případě potřeby je možné záhlaví rozšířit (stačí úprava této třídy), a to přidáním dalšího parametru do konstruktoru třídy a úpravou formátovacího řetězce pro načítání/ukládání dat záhlaví do posloupnosti bajtů (např. zde popsané záhlaví má formátovací řetězec `<32sB?BI`, který reprezentuje řetězec (*string*) uložený ve 32 B v pořadí little endian (haš), dále 1B číselnou hodnotu typu *byte* (typ rozprostření), 1B hodnotu typu *bool* (příznak komprimace), 1B hodnotu typu *byte* (hloubka modifikace) a 4B hodnotu typu *unsigned int* (délka skrývaných dat).

5.4.2 Skrývání dat

Skrývání dat realizuje třída `StegoOut`. Mimo různých pomocných metod implementuje metody pro zajištění načtení vstupního souboru WAV (nosiče), souboru se skrývanými daty, pro uložení souboru WAV se skrytými daty, pro nastavení parametrů, vlastní spuštění procesu skrytí (vč. případné komprese nebo šifrování, přičemž jednotlivé metody pro skrývání se dělí podle způsobu rozprostření):

- `wav_src_load` – načtení vstupního souboru WAV
- `data_src_load` – načtení souboru s daty pro skrytí
- `setparams` – nastavení parametrů skrývání (hloubka modifikace, způsob rozprostření, heslo, aktivace komprese, šifrování)
- `wav_tgt_write` – uložení souboru WAV se skrytými daty
- `process` – spuštění skrytí dat na základě načtených vstupů a nastavených parametrů
- `reset` – vymazání obsahu zvukového záznamu se skrytými daty (v rámci instance třídy)
- `check_input` – ověření, zda lze pro načtené vstupy (WAV soubor, soubor s daty pro skrytí) provést skrytí dat, tj. jestli je steganografická kapacita nosiče dostatečná, a za jakých podmínek (s jakou minimální hloubkou modifikace a s jakým stavem komprimace); tato metoda se volá externě po načtení vstupních dat
- `stop` – zastavení procesu skrývání (skrývání se v programu spouští v novém vlákně z důvodu potenciální časové náročnosti)

Při skrývání dat se mj. využívá třídy `Bitstream`, která slouží pro vytvoření seznamu (*list*) s bajty obsahujícími na nejnižších bitových pozicích n bitů ze souboru s tajnými daty. Bajty z tohoto seznamu se poté využívají k nahrazení nejnižších bitů v bajtech vzorků při procesu skrývání. Třída `Bitstream` implementuje tyto metody:

- `get_bytearray` – vrací seznam bajtů, které obsahují vždy po řadě n bitů (na pozicích LSB) ze vstupních dat (souboru s daty pro skrytí)
- `stop` – zastaví generování seznamu bajtů obsahujících na pozicích LSB bity ze vstupních dat (zastavení se provede, pokud uživatel zruší proces skrývání dat)

V programu Stego se po patřičném příkazu z GUI pro načtení vstupních dat (z menu) načte soubor WAV, resp. soubor s daty pro skrytí, a po tomto načtení se spustí kontrola na proveditelnost skrytí dat. Na základě výsledku kontroly se v případě nemožnosti skrytí zobrazí uživateli chybové hlášení, popř. informace o tom, že je nezbytné provést kompresi. Následně se v případě proveditelnosti nastaví v prvku pro volbu modifikace minimální možná hodnota a případně se i aktivuje komprimace dat (a znemožní se její vypnutí pomocí patřičného zaškrtávacího pole).

Po spuštění skrývání dat uživatelem (z GUI) se zobrazí informační dialog (`ProcessDialog`). Ten spustí skrývání dat v novém vlákně (aby nebylo zablokováno ovládání GUI) a zobrazuje průběh skrývání. V případě potřeby může uživatel tlačítkem proces skrývání zastavit. Po dokončení procesu jsou nově vytvořená data – záznam ve formátu WAV – uložena do dočasného souboru (ve výchozím nastavení se jedná o soubor `tmp_out.wav`). Rovněž jsou nově vytvořené vzorky zobrazeny v grafu průběhu zvukového signálu.

Celý proces skrývání dat probíhá následujícím způsobem:

- Na základě podnětu uživatele proběhne načtení vstupního souboru WAV:
  - Ze vstupního souboru se načtou vlastní zvuková data (vzorky), a to za pomoci třídy jazyka Python `wave` sloužící pro práci se soubory WAV.
  - Ze vstupního souboru se načte záhlaví zvukových dat (za záhlaví se považují veškerá data, která předchází zvukovým datům).
  - Ze vstupního souboru se načte zápatí zvukových dat (tj. vše, co následuje po zvukových datech).
- Na základě podnětu uživatele proběhne načtení souboru s daty pro skrytí.
- Provede se kontrola na proveditelnost skrytí dat. Pokud kontrola neproběhne v pořádku, dojde k načtení dříve použitého souboru WAV a souboru s daty pro skrytí, pokud již tyto vstupy byly dříve úspěšně načteny.[1]
- Uživatel nastaví parametry a spustí proces skrytí dat (zobrazí se dialog pro zobrazení průběhu skrývání dat a proces skrývání se spustí v novém vlákně):
  - Pokud je aktivována komprese dat, data se zkomprimují.
  - Pokud je aktivováno šifrování dat, dojde k zašifrování dat šifrou AES-256.
  - Dojde k zápisu záhlaví skrývaných dat do zvukových vzorků, a to s hloubkou modifikace 1 LSB na vzorek. Pokud je aktivováno šifrování, záhlaví

[1] Kontrola probíhá jak při načtení souboru WAV, tak při načtení souboru s daty pro skrytí. Tyto vstupy mohou být načteny v libovolném pořadí.

se před vložením do vzorků zašifruje.
- Provede se skrytí dat dle nastaveného druhu rozprostření.
- Zapíše se případné zápatí souboru WAV.
- Veškerá nově vytvořená data (WAV záhlaví + upravené vzorky + WAV zápatí) se uloží do dočasného souboru.

5.4.3 Extrakce dat

Extrakce dat je prováděna za pomoci třídy `StegoIn`. Samotná extrakce je spuštěna příkazem z nabídky (menu), přičemž nejdříve je pomocí dialogu pro otevření souboru stanovena cesta k souboru WAV se skrytými daty, a následně je spuštěna extrakce (v novém vlákně, podobně jako u procesu skrytí dat). Při extrakci dojde k načtení vzorků ze souboru, načtení obsahu záhlaví skrytých dat, a následně se spustí první stupeň kontroly dat.

V prvním stupni kontroly se zjišťuje smysluplnost obsahu záhlaví, tedy

- zda není nastavena neplatná metoda rozprostření,
- zda není nastavena neplatná hloubka modifikace,
- zda není uvedena neplatná délka uložených dat.

Touto kontrolou lze hned na začátku procesu extrakce dat zabránit nekorektnímu načtení souborů, které jsou poškozené, neobsahují skrytá data nebo může být jejich obsah šifrovaný. Pokud je obsah záhlaví neplatný, je na tuto skutečnost uživatel upozorněn a pokud ví, že soubor skrytá data obsahuje a data jsou šifrovaná, může zadat heslo a znovu obsah souboru načíst.

Druhý stupeň kontroly se provede v případě, že první stupeň kontroly skončí pozitivně. Ve druhém stupni se provede extrakce skrytých dat, vypočítá se jejich haš a porovná se s hašem uvedeným v záhlaví. Pokud si haše odpovídají, uživateli se zobrazí informace o souboru WAV a v něm skrytých datech. Pokud si neodpovídají, je uživatel na tuto skutečnost opět upozorněn. Následně může uživatel (pokud obě kontroly skončily úspěšně) extrahovaná data uložit do souboru dle vlastní volby.

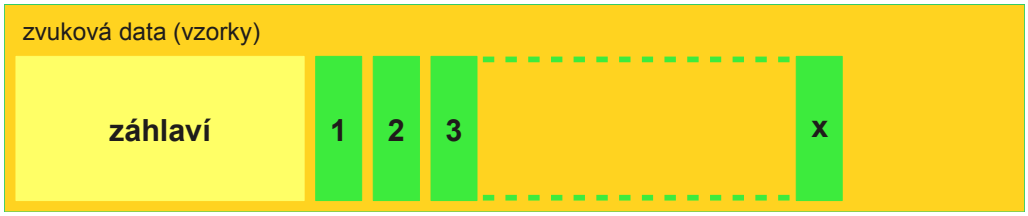
5.4.4 Způsoby rozprostření

Sekvenční rozprostření

Program Stego pracuje se třemi způsoby rozprostření, tedy způsoby volby vzorků ze souboru WAV, do jejichž LSB se skryjí tajná data. Výchozím způsobem je **sekvenční rozprostření**[2], při kterém jsou měněny LSB jednotlivých vzorků v jejich původním

[2]Striktně řečeno se nejedná přímo o rozprostření, avšak tento termín je pro zachování konzistence s ostatními způsoby vkládání skrývaných dat použit i zde

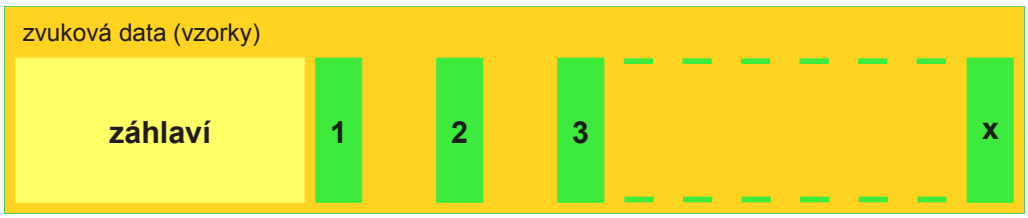
pořadí (viz obr. 5.4). Vždy se načte původní vzorek, načte se bajt (resp. 2 bajty pro 16bitové vzorky) ze seznamu vytvořeného metodou `get_bytearray` třídy `Bitstream` a n nejméně významnými bity z načtených bajtů se přepíše n LSB ve vzorku. Jakmile dojde k zapsání všech potřebných dat, doplní se zbývající, neupravené vzorky z původního souboru.



Obr. 5.4: Sekvenční rozprostření dat do vzorků.

Rovnoměrné rozprostření

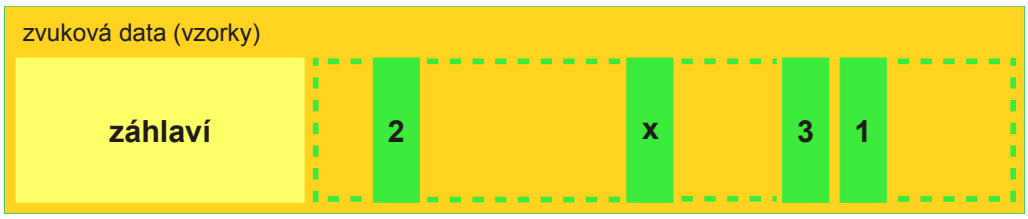
Při rovnoměrném rozprostření se na základě hloubky modifikace, délky skrývaných dat a počtu dostupných vzorků vypočte krok, se kterým se budou vybírat vzorky pro skrytí dat, a to s cílem pokrýt rovnoměrně celou posloupnost vzorků zvukového souboru, jak je znázorněno na obr. 5.5. Nejdříve se načtou do nového pole (*bytearray*) všechny původní vzorky, které budou dále na patřičných pozicích v poli upravovány. Začíná se vždy prvním dostupným vzorkem (tj. vzorkem, který následuje za posledním vzorkem, do kterého byla ukryta data záhlaví), stejně jako u ostatních způsobů rozprostření. Následně se k pozici prvního vzorku v poli přičte vypočtený krok (počet přeskočených vzorků) a vzhledem k tomu, že tento krok nemusí být celé číslo, se vypočtená pozice zaokrouhlí na celé číslo (aby mohla identifikovat vzorek v poli). Do vzorku na této pozici se zapíše požadovaný počet LSB, k pozici se opět přičte krok a výsledná hodnota se zaokrouhlí, do vzorku na nové pozici se opět vloží daný počet LSB, a tento proces se opakuje do doby, než jsou zapsány všechny bity tajných dat.



Obr. 5.5: Rovnoměrné rozprostření dat do vzorků.

Pseudonáhodné rozprostření

Tento způsob rozprostření je založen na pseudonáhodném výběru pozic v poli bajtů reprezentujících zvukové vzorky, do kterých se budou vkládat bity tajných dat. Mimo vlastní skrývaná data je nutným vstupem i heslo, které určuje, jakým způsobem se pokryje cílový prostor všech vzorků. Princip (dle diagramu na obr. 5.7) je následující: Na základě hesla je vypočten haš (funkcí SHA2-256), ze kterého se využije prvních 8 bajtů (které jsou převedeny na celé číslo), a toto číslo je vyděleno operací *dělení modulo* počtem vzorků. K výslednému zbytku po dělení se ještě přičte *offset* (hodnota odpovídající počtu vzorků, které už byly obsazeny skrytím záhlaví), a tato výsledná hodnota jednoznačně v rámci všech vzorků identifikuje pořadí vzorku, do kterého se zapíší bity tajných dat. Pro další vzorky se postupuje podobným způsobem, ale jako vstup hašovací funkce již neslouží heslo, ale předchozí vypočtená hodnota haše.



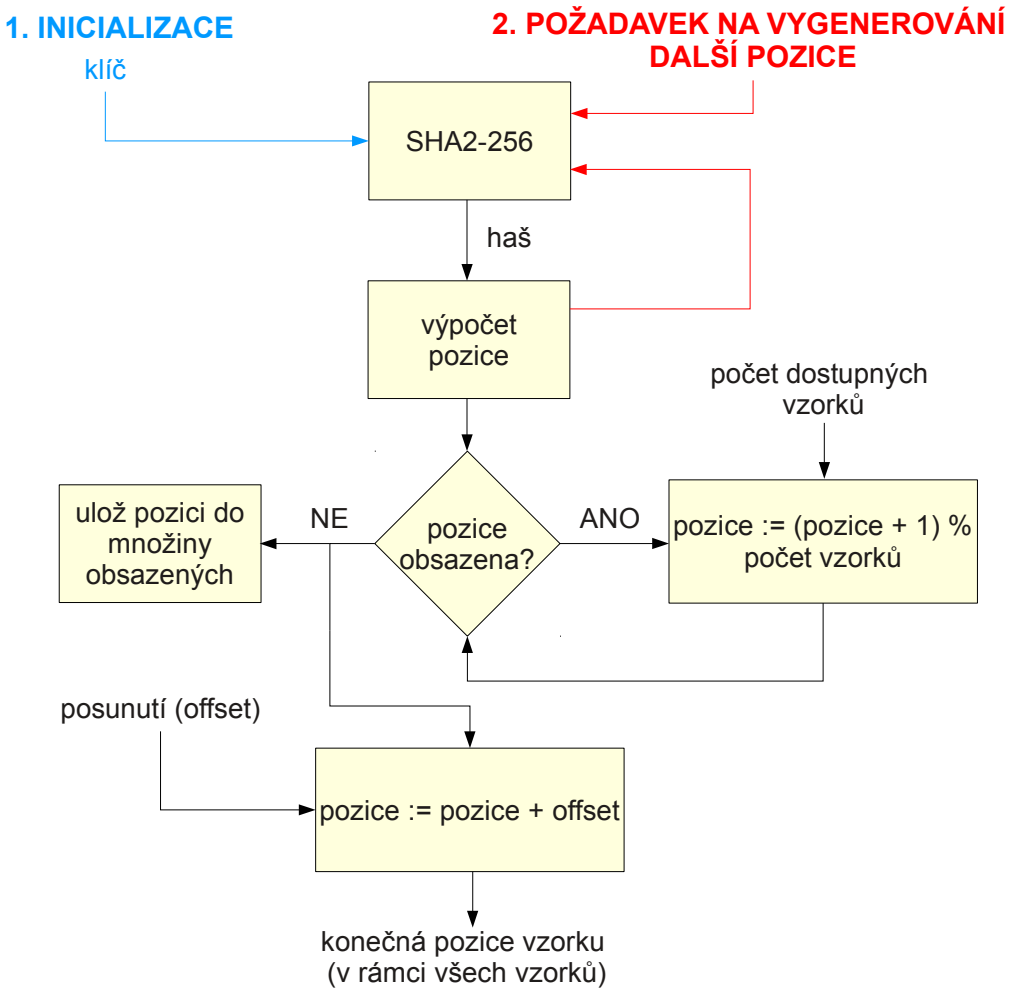
Obr. 5.6: Pseudonáhodné rozprostření dat do vzorků.

Protože se může stát, že se během výpočtů jednotlivých pozic vypočtou hodnoty, které byly již vypočteny dříve, jsou veškeré již vypočtené hodnoty ukládány do množiny obsazených pozic. Pokud se vypočte stejná hodnota, je nalezena hodnota první volné pozice (modulo počet vzorků), která následuje za vypočtenou obsazenou pozicí. Výsledné rozprostření dat do jednotlivých vzorků přibližuje obr. 5.6.

Z implementačního hlediska je k výpočtu pozic použit *generátor* – konstrukce jazyka Python, která po inicializaci vytvoří objekt iterativně generující definovaným způsobem výstupní hodnoty, pokud je zavolána funkce `next` a generátor je uveden jako její parametr. V tomto případě se jedná o generátor `position_generator`, kterému je při inicializaci předáno jako parametr heslo (libovolný hašovatelný datový typ), počet vzorků a počet obsazených vzorků (offset). Následně jsou pozice jednotlivých vzorků, do kterých se budou skrývat data, postupně získávány voláním metody `next`, které je objekt generátoru předán jako parametr.

5.4.5 Šifrování

K volitelnému šifrování dat je využito třídy `AES` z modulu `PyCrypto` (zdrojový kód pro šifrování dat byl inspirován kódem z [18]). V programu Stego šifrování a



Obr. 5.7: Princip generování pozic vzorků pro pseudonáhodné rozprostření

dešifrování realizuje třída `AES256` prostřednictvím metod `encrypt` a `decrypt`, které mají jako vstupní parametry heslo nebo šifrovací klíč, data k zašifrování/dešifrování a volitelně délku bloku pro zpracování dat.

Při šifrování je do výstupní sekvence bajtů (pole bajtů) na její začátek uložena délka šifrovaných dat (4 B, little endian) a inicializační vektor (16 B). Dále se po blocích provádí šifrování a zašifrovaný blok dat se vždy přidá na konec výstupní sekvence. Při dešifrování probíhá inverzní postup.

5.4.6 Přehrávání zvukových záznamů

Přehrávání zvukových záznamů typu WAV zajišťuje třída `WavPlayer`, která využívá třídy `pyaudio`[14] ze stejnojmenného modulu jazyka Python. Cesta k přehrávanému souboru se předává při inicializaci třídy. Přehrávání je realizováno v novém vlákně, aby neblokovalo ovládací prvky rozhraní GUI a bylo možné přehrávání zastavit

(tj. po opětovném spuštění se začne přehrávat od začátku) nebo pozastavit (po opětovném spuštění přehrávání pokračuje od místa, ve kterém bylo pozastaveno). Přehrávání se řídí metodami

- `play` pro spuštění přehrávání,
- `pause` pro pozastavení,
- `stop` pro zastavení a
- `toggle` pro změnu stavu, tj. přehrávání/pozastavení.

5.4.7 Grafické uživatelské rozhraní

Grafické rozhraní je zajišťováno především třídou `StegoApp`. Po inicializaci této třídy dojde k vytvoření hlavního okna programu a rozmístění ovládacích prvků. Veškeré činnosti (akce) se spouští metodami této třídy, které jsou volány aktivací odpovídajícího ovládacího prvku, a samy případně dále volají pomocné obslužné metody.

V části pro GUI jsou mimo inicializace akcí (načtení vstupních dat, skrytí dat do souboru WAV, vizualizace) rovněž implementovány kontroly a ošetření vstupů. Jejich cílem je zajistit, aby uživatel nemohl aktivovat činnost, která je v daném stavu nekorektní. Jedná se především o nastavení minimální a maximální hloubky modifikace vzorků, zamezení spuštění skrývání dat, pokud nebyly načteny potřebné vstupy (soubor WAV a skrývaná data) nebo bylo nastaveno pseudonáhodné rozprostření, ale nebylo zadáno heslo.

Kromě hlavního okna jsou v grafickém rozhraní implementována dílčí okna (dialogy) se specifickými funkcemi. Tyto dialogy jsou založeny na třídě `SimpleDialog`[16], kterou dále rozšiřují dle konkrétní funkce. Prvním dialogem je `ExtractDialog`, který je využíván při extrakci dat ze souboru WAV. Zobrazují se v něm informace o zvukovém záznamu a skrytých datech, umožňuje zadat heslo (pokud jsou data šifrována nebo je použita pseudonáhodná metoda rozprostření) a uložit tajná data do souboru. Dalším dialogem je `PasswordDialog`, který slouží k zadávání hesla při šifrování nebo pseudonáhodném rozprostření (heslo se zadává dvakrát a kontroluje se shoda obou hesel, aby nedošlo k překlepu).

Během skrývání nebo extrakce dat je využito dialogu `ProcessDialog`. Jeho účelem je spustit proces skrývání nebo extrakce v novém vlákně a dotazovat se periodicky na procentuální průběh dané operace a zobrazovat aktuální procentuální hodnotu průběhu, aby si uživatel mohl udělat představu o tom, jaká část zpracování je již hotova a jaká ještě zbývá. Rovněž poskytuje uživateli možnost tento proces přerušit, např. pokud si uživatel uvědomí, že zadal nevhodné parametry nebo usoudí, že by zpracování trvalo dlouho.

K vizualizaci dat je mimo grafického průběhu zvukových signálů využit i dialog `HexViewDialog`. Jsou v něm zobrazena data (bajty) z původního i nově vytvořeného

souboru WAV v textové podobě a v podobě šestnáctkových čísel (bližší popis v sekci 5.4.8). Posledním dialogem je okno s informacemi o programu – `AboutDialog`.

5.4.8 Vizualizace dat

Vizualizace dat je provedena dvěma způsoby. Prvním způsobem je grafické zobrazení průběhu zvukového signálu za pomoci prvku GUI `WavePlot`, který sestává především z komponenty `canvas` z knihovny `matplotlib` pro vykreslování grafů a dále z komponenty (třídy) pro zobrazení legendy grafu `Legend`, panelu s tlačítky pro práci s grafem `PlotToolbar` a zaškrtávacího přepínače pro zobrazení mřížky. Zobrazení je provedeno jak pro vzorky z původního souboru WAV, tak i pro vzorky z nově vytvořeného souboru se skrytými daty. Zvukový signál je v grafu zobrazen ihned po načtení vstupního souboru, resp. je doplněn zobrazením zvukového signálu souboru se skrytými daty poté, co je provedeno skrytí dat.

Oba průběhy jsou barevně odlišeny, barvy a míra průhlednosti jsou nastavitelné v konfiguračním souboru. Tlačítky u grafu lze zapnout funkce pro práci s grafem – posun zobrazované oblasti (resp. snižování/zvyšování měřítka zobrazení) a přiblížení nebo oddálení vybrané obdélníkové oblasti v grafu. Zobrazení vzorků v grafu je rozděleno do dvou úrovní podle úrovně přiblížení. Na první úrovni je načtena posloupnost vzorků, která je tvořena původní posloupností vzorků omezenou podvzorkováním. Snížení počtu načítaných vzorků je nezbytné jednak z hlediska rychlosti zpracování a také z důvodu paměťové náročnosti komponenty `canvas`. Pokud uživatel přiblíží oblast grafu tak, že šířka zobrazované oblasti v pixelech vynásobená konstantou `SAMP_RESERVE` (její hodnota je nastavena na 100) je menší než počet zobrazovaných vzorků (po podvzorkování), jsou do komponenty pro zobrazování grafu načteny vzorky pokrývající zobrazovanou oblast, nyní již bez podvzorkování. Tímto přístupem je snížena doba odezvy při práci s grafem a paměťová náročnost, avšak při větším přiblížení je zachována přesnost zobrazení.

Druhou možností zobrazení dat je použití dialogu `HexViewDialog`, jehož stěžejní součástí je komponenta `HexView`. Jsou v něm zobrazena kompletní data původního i nově vytvořeného souboru WAV (tedy na rozdíl od grafu nikoliv jen de facto hodnoty vzorků) po jednotlivých bajtech. Bajty jsou zobrazeny jednak v textové formě a jednak ve formě odpovídajících šestnáctkových čísel, a to ve čtyřech sloupcích (viz obr.A.3). Rovněž je volitelně zobrazena pozice odpovídajících řádků v rámci souboru (pozice začátku daného řádku v bajtech od začátku souboru). Bajty, které se vzájemně v původním a novém souboru liší, jsou barevně zvýrazněny (barva textu i pozadí je konfigurovatelná). Uživatel si rovněž může myší označit určitou oblast, přičemž se označí i odpovídající data v ostatních sloupcích, což může posloužit k lepší orientaci.

Komponenta (třída) `HexView` je implementována za využití pěti textových polí `ReadonlyText`[3]. Obsah obou souborů je načten do polí (*bytearray*), která jsou atributy třídy `HexView`, ale z důvodu paměťové náročnosti komponenty `Text` není do patřičných textových polí načten celý obsah souborů. Načtena je vždy pouze část dat, která odpovídá zobrazované oblasti, a při události (změna pozice vertikálního posuvníku, otočení kolečka myši, stisk klávesy PageUp/PageDown/šipka dolů/šipka nahoru nebo změna velikosti komponenty `HexView`) se přepočítá, jaká část dat se má zobrazit, a tato data se zobrazí a zvýrazní se v nich případné odlišnosti.

5.4.9 Nápověda

Nápověda k programu je pojata ve formě Uživatelské příručky. Protože sada komponent Tkinter neobsahuje komponentu vhodnou pro jednoduché a flexibilní zobrazení netriviálně formátovaného textu s obrázky a dalšími náležitostmi, byla nápověda vytvořena ve formě souboru HTML (s požadovaným formátováním za pomoci kaskádových stylů – CSS). Po vyvolání nápovědy z programu dojde k otevření patřičného souboru (`help.html`) ve výchozím internetovém prohlížeči.

Uživatelská příručka obsahuje detailní popis programu Stego z uživatelského hlediska, a práce s ním. Rovněž se zaměřuje i na popis možností konfigurace programu a stručně zmiňuje i prostředky použité při vývoji programu (knihovny, programovací jazyk, nástroj `cx_freeze`).

5.4.10 Konfigurace

Některé z vlastností programu jsou konfigurovatelné pomocí konfiguračního souboru. Jedná se o konfigurační soubor typu `ini`, který je výhodný svou jednoduchou strukturou a je podporován v základní instalaci jazyka Python za pomoci modulu `configparser`. Struktura souboru je tvořena jednotlivými oddíly (*sekcemi*), které logicky rozčleňují konfigurační parametry podle oblasti jejich použití; názvy sekcí jsou ohraničené hranatými závorkami. Nastavení konfiguračních voleb se zapisuje ve formátu `parametr = hodnota`, komentáře nebo popisky jsou uvozeny středníkem.

Nastavování konfiguračních parametrů využívá funkci `cfgval`, která byla vytvořena, aby bylo možné zároveň s načtením konfiguračních parametrů brát v úvahu s nimi spojená specifika. Především je důležité, aby program fungoval i ve stavu, kdy neexistuje konfigurační soubor nebo v něm není některý z parametrů uveden. V takovém případě funkce `cfgval` použije hodnotu zadanou konstantou z kódu, která definuje výchozí hodnotu (tj. tato hodnota parametru se použije, pokud není

[3]Jedná se o úpravu komponenty `Text`, která sama o sobě neumožňuje nastavit, zda bude v ní obsažený text nezměnitelný (jen pro čtení). Kód komponenty `ReadonlyText` byl převzat z [17]

explicitně v konfiguračním souboru stanoveno jinak). Dalším specifikem je, že při načtení hodnoty parametru je tato hodnota typu `str` (řetězec), takže v některých případech je nutné ji převést na celé nebo desetinné číslo.

Pomocí parametrů lze nastavit některé funkční vlastnosti programu (např. koeficient podvzorkování při vykreslování průběhu zvukového signálu), a také vlastnosti vizuální. Jedná se zejména o barvy (průběh zvukových signálů, fonty, pozadí), zobrazení mřížky v komponentě pro vykreslování grafů, nebo o rodinu fontu a velikost písma v případě HEX prohlížeče. Popis jednotlivých parametrů je uveden v uživatelské příručce (příloha A).

6 ZÁVĚR

Steganografie nabízí mnoho různých možností utajení uložených či přenášených dat, přičemž jednou z možných oblastí aplikací steganografie je skrývání tajných dat do zvukových záznamů. I v této oblasti existuje mnoho různých metod a technik, jak data do záznamu skrýt.

Na základě popsaných a prozkoumaných možností ukrývání dat ve zvukových záznamech byla za účelem návrhu laboratorní úlohy zvolena technika LSB, která je založena na úpravě nejméně významných bitů ve vzorcích zvuku. Dále bylo popsáno, jakým způsobem se vzorky upravují a jakou strukturu mají soubory typu WAV, které budou využity jako nosiče pro steganografickou techniku LSB.

Kombinace techniky LSB a souborů WAV byla vybrána proto, že dané mechanismy a způsob ukládání dat je možné poměrně jednoduše pochopit a na základě toho přímočaře ilustrovat jeden z možných přístupů při skrývání dat. Na těchto základech byla navržena laboratorní úloha, ve které si studenti budou moci vyzkoušet aplikaci steganografie ve zvukových záznamech a vliv různých nastavení a různého množství a typu dat na výsledný zvukový záznam (zejména vliv hloubky modifikace a vliv šifrování). Zadání úlohy bylo navrženo s důrazem na klíčové body steganografie, a to jak z teoretického, tak i z praktického hlediska. Úloha má rovněž za cíl podnítit uvažování studentů a přimět je, aby se nad jevy, se kterými se setkají, zamysleli a odhalili patřičné souvislosti a důsledky.

Pro laboratorní úlohu byl navržen počítačový program Stego, který tvoří nedílnou součást laboratorní úlohy a studentům poskytuje nástroj pro ověření a prozkoumání teoretických předpokladů v oblasti steganografie ve zvukových záznamech. S ohledem na využití v laboratorní úloze byla přizpůsobena i stránka ovládání a možnosti nastavení programu.

Lze předpokládat, že po praktickém nasazení programu a laboratorní úlohy do výuky bude vhodné provést částečné úpravy (ať již v zadání úlohy nebo ve vstupních datech) tak, aby ještě lépe odrážely konkrétní potřeby výuky na základě konkrétních zkušeností studentů a vedoucích cvičení.

Program rovněž poskytuje prostor pro další rozšíření, například formou navazující studentské práce. Možností k funkčnímu doplnění by bylo přidání dalších způsobů skrytí dat či zdokonalení stávajících (například možnost ovlivnění LSB ve vzorcích, které nenesou tajná data, možnost skrytí dat do vzorků na náhodných pozicích, přičemž pro extrakci tajných dat by bylo nutné mít k dispozici i původní zvukový záznam, apod.). Další možnosti rozšíření by bylo možné nalézt ve způsobu vizualizace a reprezentace dat (například přidat zobrazení kmitočtového spektra zvukového signálu, zvýraznit rozdílné části zvukového záznamu v závislosti na míře odlišnosti, aj.).

LITERATURA

[1] MORKUS, Filip. *Program pro skrývání dat v obrazových souborech.* Brno, 2011. Dostupné z: <https://dspace.vutbr.cz/bitstream/handle/11012/5152/DiplomovaPrace_Morkus_88811.pdf?sequence=1>. Diplomová práce. Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Karel Burda.

[2] ŽILKA, Roman. *Steganografie a stegoanalýza* [online]. 2008 [cit. 2013-12-31]. Diplomová práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Zdeněk Říha. Dostupné z: <http://is.muni.cz/th/73058/fi_m/>.

[3] Argot. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-12-30]. Dostupné z: <http://en.wikipedia.org/wiki/Argot>

[4] Steganalysis. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-12-30]. Dostupné z: <http://en.wikipedia.org/wiki/Steganalysis>

[5] HARBARCHUK, Volodymyr a Grzegorz KOZIEL. The Review of Sound-based Steganographic Techniques. *Iskustvennyj Intellekt.* 2008, roč. 2008, č. 4. Dostupné z: <http://archive.nbuv.gov.ua/portal/natural/ii/2008_4/JournalAI_2008_4/Razdel1/01_Harbarchuk.pdf>

[6] White noise. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-12-30]. Dostupné z: <http://en.wikipedia.org/wiki/White_noise>

[7] WAV. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-12-30]. Dostupné z: <http://en.wikipedia.org/wiki/Wav>

[8] Linear pulse-code modulation. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-12-30]. Dostupné z: <http://en.wikipedia.org/wiki/Linear_pulse-code_modulation>

[9] SAPP, Craig Stuart a Scott WILSON. WAVE PCM soundfile format. Music 422: Perceptual Audio Coding: Projects [online]. 2003 [cit. 2013-12-31]. Dostupné z: <https://ccrma.stanford.edu/courses/422/projects/WaveFormat/>

[10] Resource Interchange File Format. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-12-31]. Dostupné z: <http://en.wikipedia.org/wiki/Resource_Interchange_File_Format>

[11] PYTHON SOFTWARE FOUNDATION. *Python 3.3.5 documentation* [online]. 2014, 2014-03-18 [cit. 2014-05-01].
Dostupné z: <https://docs.python.org/3.3/>

[12] JAVA2S.COM. *GUI Tk Python*> [online]. 2012 [cit. 2014-05-01]. Dostupné z: <http://www.java2s.com/Code/Python/GUI-Tk/CatalogGUI-Tk.htm>

[13] LUNDH, Fredrik. *Effbot.org: An Introduction to Tkinter (Work in Progress)* [online]. 2013 [cit. 2014-05-01].
Dostupné z: <http://effbot.org/tkinterbook/>

[14] PHAM, Hubert. PyAudio Documentation [online]. 2014 [cit. 2014-05-01]. Dostupné z: <http://people.csail.mit.edu/hubert/pyaudio/docs/>

[15] THE MATPLOTLIB DEVELOPMENT TEAM. *Overview – Matplotlib 1.3.1 documentation* [online]. 2013, 2013-10-10 [cit. 2014-05-01].
Dostupné z: <http://matplotlib.org/contents.html>

[16] LUNDH, Fredrik. Dialog Windows. *Effbot.org* [online]. 2013 [cit. 2014-05-01]. Dostupné z: <http://effbot.org/tkinterbook/tkinter-dialog-windows.htm>

[17] ReadOnlyText. *Tkinter Wiki* [online]. 2010, 2010-07-26 [cit. 2014-05-01]. Dostupné z: <http://tkinter.unpythonic.net/wiki/ReadOnlyText>

[18] BENDERSKY, Eli. AES encryption of files in Python with PyCrypto. In: *Eli Bendersky's website* [online]. 2010, 2013-11-15 [cit. 2014-05-01]. Dostupné z: <http://eli.thegreenplace.net/2010/06/25/aes-encryption-of-files-in-python-with-pycrypto/>

[19] TUININGA, Anthony. *Cx_Freeze 4.3.3 documentation* [online]. 2013 [cit. 2014-05-01]. Dostupné z: <http://cx-freeze.readthedocs.org/en/latest/>

SEZNAM ZKRATEK

AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
BE	Big Endian
LSB	Least Significant Bits
FLAC	Free Lossless Audio Codec
GUI	Graphical User Interface
JPEG	Joint Photographic Expert Group
JPG	Joint Photographic Expert Group
LE	Little Endian
LPCM	Linear Pulse-Coded Modulation
MAC	Message Authentication Code
MP3	MPEG Audio Layer 3
MPEG	Moving Picture Experts Group
OS	operační systém
PCM	Pulse-Coded Modulation
RIFF	Resource Interchange File Format
SHA2	Secure Hash Algorithm 2
WAV	Wave
XOR	eXclusive OR
CSS	Cascading Style Sheet
HTML	HyperText Markup Language
PDF	Portable Document File

SEZNAM PŘÍLOH

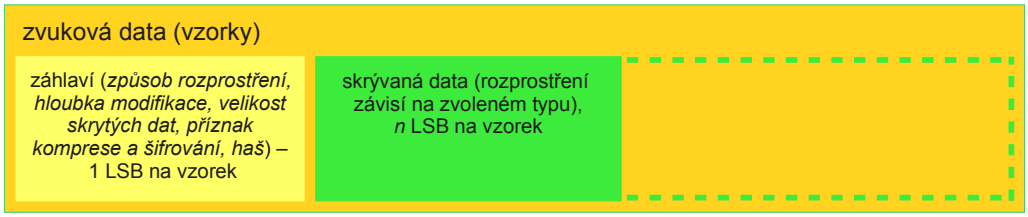
A UŽIVATELSKÁ PŘÍRUČKA PROGRAMU

V této příloze je uveden text uživatelské příručky, která je součástí programu Stego. Formátování bylo z původní podoby upraveno tak, aby lépe vyhovovalo tisku.

A.1 Popis programu

Program Stego slouží ke skrývání dat libovolného charakteru do zvukových souborů typu WAV za pomoci techniky *LSB* (Least Significant Bits, nejméně významné bity), tj. techniky, která skrývá bity skrývaných dat na místo bitů nejnižších řádů ve vzorcích zvukového záznamu. Podporovány jsou soubory s 8 a 16 bity na vzorek a jedním nebo dvěma zvukovými kanály (mono/stereo). Umožňuje uživateli zvolit si hloubku skrývání dat (tedy počet LSB, které budou ve vzorku nahrazeny adekvátně dlouhou sekvencí bitů ze skrývaných dat) a způsob rozprostření dat ve zvukovém záznamu:

- **Sekvenční rozprostření** (výchozí volba) – vzorky pro skrytí dat jsou vybírány sekvenčně od začátku dostupných vzorků zvukového záznamu.
- **Rovnoměrné rozprostření** – vzorky pro skrytí dat jsou vybírány přibližně rovnoměrně přes celou posloupnost dostupných vzorků záznamu.
- **Pseudonáhodné rozprostření** – pozice vzorků pro skrytí dat je vybírána v rámci dostupných vzorků iterativně pseudonáhodným způsobem za využití hašovací funkce SHA2-256. Pro tento způsob je nutné zadat heslo, na jehož základě jsou vypočítávány pozice upravovaných vzorků v záznamu.



Obr. A.1: Struktura zvukových vzorků

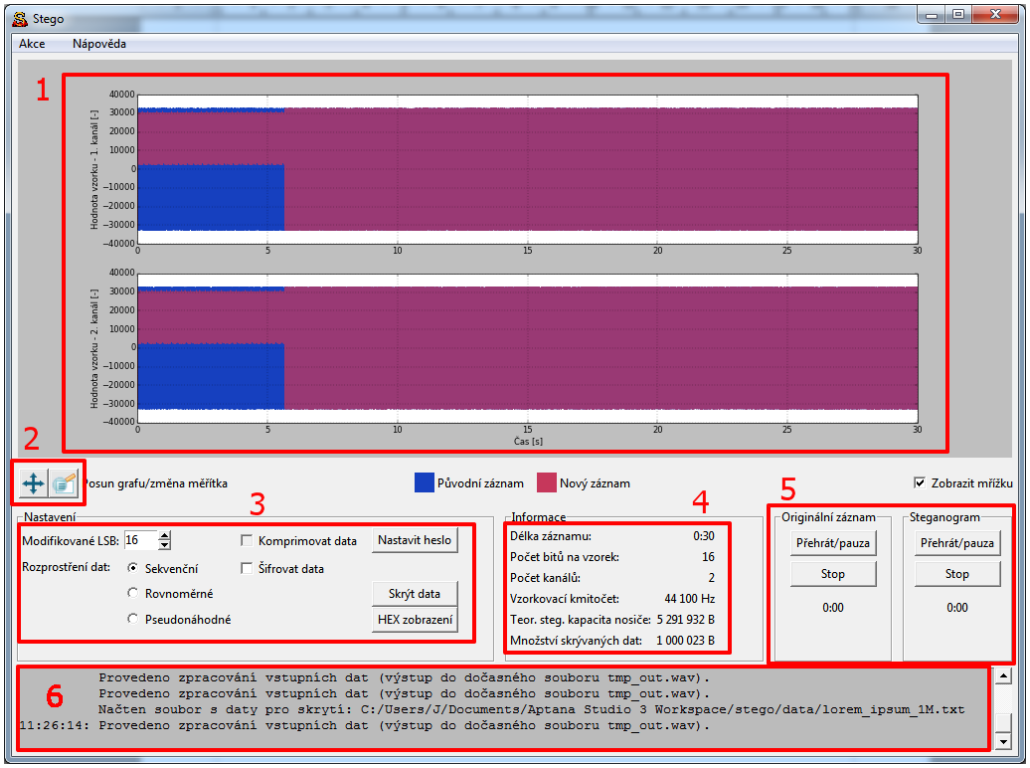
Před vlastními skrývanými daty je do zvukového záznamu na jeho počátku vloženo ještě záhlaví s informacemi o skrývaných datech (počet přepisovaných LSB, délka uložených dat, příznak komprese dat, haš dat sloužící jako kontrolní součet). Tyto informace jsou do vzorků ukládány s hloubkou modifikace 1 LSB na vzorek. Dostupné vzorky pro ukládání skrývaných dat tedy následují až za vzorky, ve kterých je uloženo záhlaví (viz obr. A.1).

Program umožňuje volitelnou kompresi skrývaných dat typu Bzip2 (což v některých případech dovoluje uložit do záznamu stejné množství dat s menším poškozením zvukového záznamu, resp. při stejném poškození uložit větší množství dat). Skrývaná data je možné zašifrovat (šifrování se provádí včetně záhlaví) blokovou šifrou AES-256. Mimo skrývání dat samozřejmě program disponuje i funkcí pro extrakci skrytých dat z WAV souboru (steganogramu).

Původní zvukový záznam i nově vytvořený záznam (steganogram) jsou reprezentovány ve formě grafu průběhu zvukového signálu v čase (resp. v rámci sekvence vzorků) – volba typu časové osy je nastavitelná v konfiguračním souboru. Další možnost reprezentace je zobrazení původního a nového souboru jako binárních dat po bajtech.

A.2 Ovládání programu

A.2.1 Hlavní okno



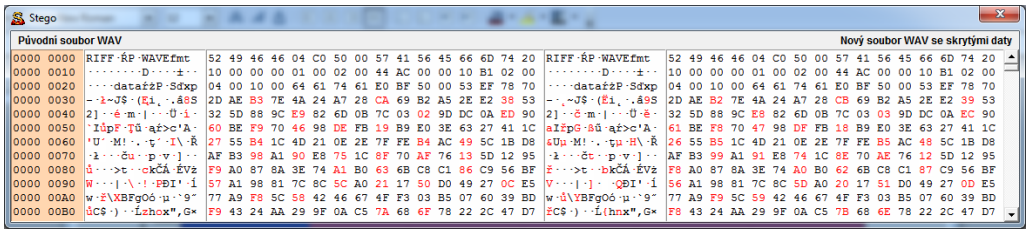
Obr. A.2: Hlavní okno programu Stego

Hlavní okno programu (obr. A.2) sestává z těchto hlavních celků:

1. **Grafická reprezentace průběhu zvukového signálu** (hodnot vzorků v sekvenci vzorků, resp. v čase). Zobrazenou oblast je možné přibližovat, oddalovat

a posunovat (blíže viz následující bod). Pro lepší orientaci je ve výchozím stavu zobrazena mřížka, je možné ji vypnout přepínačem na pravé straně okna pod oblastí grafu. Výchozí stav zobrazení mřížky je konfigurovatelný.

2. **Tlačítka pro úpravu zobrazení grafu.** Prvním tlačítkem se přepíná na funkci posunu/změny měřítka. Při pohybu myší v oblasti grafu se stiskutým levým tlačítkem dochází k posunu zobrazené části signálu. Při pohybu se stisknutým pravým tlačítkem se mění měřítko zobrazení. Druhým tlačítkem se zapíná funkce přibližování/oddalování vybrané obdélníkové oblasti (přibližování kliknutím a tažením levým tlačítkem myši, oddalování tlačítkem pravým).
3. **Nastavení hloubky modifikace** (počtu LSB), typu rozprostření, aktivace komprese dat nebo šifrování. Pro šifrování a/nebo volbu pseudonáhodného rozprostření je nutné zadat heslo (po kliknutí na tlačítko *Nastavit heslo*). Za předpokladu, že byl načten vstupní zvukový soubor (nosič) a data určená pro skrývání, a steganografická kapacita nosiče je pro uložení dat dostatečná, je možné provést skrytí dat tlačítkem *Skrýt data*. Poté, co proběhlo skrytí dat, je možné zobrazit porovnání vstupního a výstupního souboru ve formě posloupnosti bajtů a jejich hexadecimální reprezentace (obr. A.3) kliknutím na tlačítko *HEX zobrazení*.
4. **Informace.** V této oblasti se zobrazují tyto informace:
   - **délka zvukového záznamu** ve formátu *mm:ss*,
   - **počet bitů na vzorek** záznamu,
   - **počet kanálů** záznamu,
   - **vzorkovací kmitočet** záznamu (=počet přehrávaných vzorků za sekundu) v kHz
   - **teoretická steganografická kapacita** nosiče – za předpokladu přepsání všech vzorků v nosiči a využití maximální hloubky skrývání dat; při zapnuté kompresi dat je možné uložit (dle typu dat) i větší množství dat, než je uvedená hodnota steg. kapacity (protože se kompresí skutečně vkládané množství dat sníží),
   - **množství skrývaných dat** (v bajtech) před případnou kompresí dat.
5. **Přehrávač** původního a nově vytvořeného zvukového záznamu. Umožňuje přehrát a zastavit nebo pozastavit zvukové záznamy a sluchově porovnat, jaký vliv mají skrytá data na kvalitu vzniklého záznamu se skrytými daty.
6. **Textové pole se záznamem provedených akcí** (nemusí být zobrazeno – záleží na nastavení v konfiguračním souboru). Podává přehled o tom, jaké akce (načtení souborů, provedení skrytí dat aj.) byly provedeny a v jakém čase.



Obr. A.3: Prohlížeč pro porovnání původního a nově vytvořeného souboru WAV

A.2.2 Načtení vstupního WAV a datového souboru

Vlastní načtení nosiče se provede v nabídce *Akce-Načíst* vstupní WAV soubor, podobně načtení skrývaných dat v nabídce *Akce-Načíst soubor s daty pro skrytí*. Steganogram je možné uložit příkazem z nabídky *Akce-Uložit výstupní WAV soubor (steganogram)*.

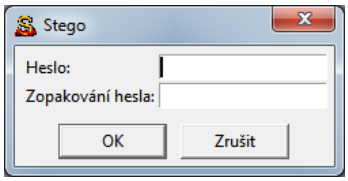
Při načítání souborů se provádí kontrola, zda je steganografická kapacita nosiče dostatečná. Pokud je tedy načten soubor WAV a následně soubor s daty pro skrytí, program zjistí, zda je pro uložení skrývaných dat steganografická kapacita WAV souboru dostatečná a pokud není, soubor s daty nenačte a zobrazí upozornění. Podobně se při načtených datech pro skrytí provádí kontrola na steg. kapacitu následně načítaného souboru WAV.

V rámci kontroly se v případě negativního výsledku dále testuje, zda by bylo možné data do souboru WAV skrýt, pokud by byla zkomprimována. Pokud je to možné, je implicitně zapnuta komprese dat a uživatel je na tuto skutečnost upozorněn. Součástí vstupní kontroly je i minimální potřebná hloubka modifikace (počet přepisovaných LSB), a na jejím základě je omezen rozsah výběru hodnot v číselném poli *Modifikované LSB*.

Pokud by docházelo k problémům z důvodu vstupní kontroly při již načtených datech nebo souboru WAV, je možné program po potvrzení uvést do výchozího stavu volbou z nabídky *Akce-Reset vstupních dat*.

A.2.3 Skrytí dat do zvukového záznamu

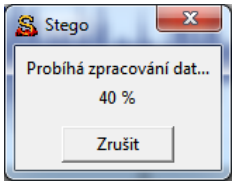
Skrytí dat je možné provést až po načtení zvukového záznamu (WAV souboru – nosiče) a dat, která budou skrývána, tlačítkem *Skrýt data*. Pokud nejsou tyto podmínky splněny, tlačítko je deaktivováno. Přepínači v části *Rozprostření dat* lze zvolit způsob rozprostření dat. Ve výchozím stavu je nastaven sekvenční způsob rozprostření. Při volbě pseudonáhodného způsobu rozprostření dat lze skrytí dat provést pouze v případě, že bylo zadáno heslo (viz dialog na obr. A.4). Vyvolání dialogu pro zadání nebo změnu hesla je možné provést stiskem tlačítka *Nastavit heslo*.



Obr. A.4: Dialog pro zadání hesla

Kompresi dat lze zapnout nebo vypnout přepínačem *Komprimovat data* (pokud nebylo použití komprese vynuceno nedostatečnou steg. kapacitou nosiče pro uložení dat v nekomprimované formě).

Šifrování dat lze aktivovat a deaktivovat přepínačem *Šifrovat data*. Při aktivovaném šifrování lze provést skrytí dat jen v případě, že bylo zadáno heslo.



Obr. A.5: Průběh zpracování dat

Po spuštění skrývání dat tlačítkem *Skrýt data* se zobrazí dialog (viz obr. A.5), který informuje uživatele o průběhu procesu skrývání dat a umožňuje proces přerušit.

A.2.4 Vizualizace

Zvukový záznam z načteného souboru WAV je zobrazen v grafu průběhu zvukového záznamu (signálu) v horní části okna. Spolu s ním je vizualizován i nově vytvořený zvukový záznam obsahující skrytá data, pokud již bylo provedeno skrytí dat. Legenda (příslušné barvy) pro zobrazení původního a nového záznamu je umístěna v oblasti pod grafem; barvy je možné nastavit v konfiguračním souboru.

Pokud bylo provedeno skrytí, může si uživatel prohlédnout a porovnat obsah původního a nově vytvořeného souboru v HEX prohlížeči (obr. A.3). V prohlížeči je zobrazen obsah souborů v textové formě (po jednotlivých bajtech) a rovněž jsou bajty zobrazeny v hexadecimální reprezentaci. Původní soubor je zobrazen v levé části prohlížeče, nově vytvořený soubor v pravé části. První sloupec obsahuje pozici odpovídajícího řádku v rámci souboru (posunutí oproti začátku souboru) ve formě hexadecimálního čísla.

Rozdílné bajty jsou od ostatních barevně odlišeny (barvu lze opět nastavit v konfiguračním souboru, stejně jako i další barvy v prohlížeči). Uživatel si může pro lepší

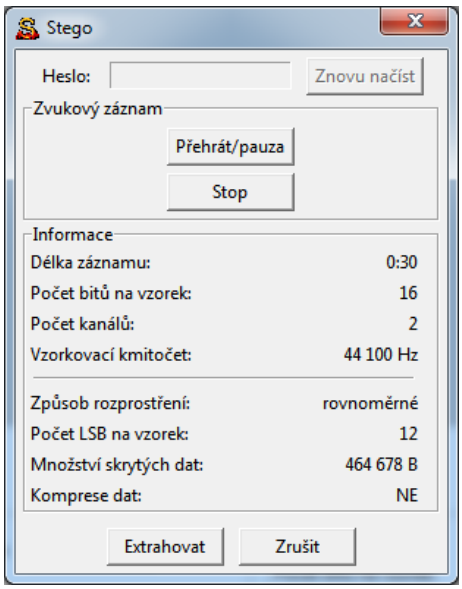
orientaci myší označit i jeden či více bajtů v libovolné části prohlížeče, což způsobí i označení odpovídajících bajtů/hexadecimálních čísel v ostatních částech prohlížeče.

A.2.5 Extrakce skrytých dat ze souboru WAV

Extrahovat skrytá data z vytvořeného steganogramu lze volbou z nabídky *Akce-Extrakce skrytých dat z WAV souboru*. Před extrakcí dat (jejíž průběh je opět signalizován dialogem jako v případě skrývání dat) se provedou dvě kontroly.

První kontrola vyhodnotí údaje v záhlaví skrytých dat (zda neobsahuje nesmyslnou délku skrytých dat, neexistující způsob rozprostření nebo nesmyslný počet LSB nesoucích skrytá data). Pokud první kontrola proběhne v pořádku, je provedena kontrola, zda se haš skrytých dat rovná haši v záhlaví. Pokud nikoliv, mohla nastat jedna z těchto možností:

1. Data jsou šifrovaná, a protože šifrování probíhá včetně záhlaví, nelze dopředu detekovat, zda se jedná o šifrování nebo některou z dalších možností.
2. Načtený soubor neobsahuje skrytá data.
3. Soubor byl poškozen či úmyslně pozměněn.



Obr. A.6: Okno pro extrakci dat

Program tuto skutečnost oznámí uživateli a vyzve jej k zadání hesla. Pokud uživatel ví, že data byla zašifrována a zná heslo, lze provést extrakci skrytých dat (po stisknutí tlačítka *Znovu načíst*).

Pokud je v souboru se skrytými daty detekována pseudonáhodná metoda rozprostření, je uživatel vyzván k zadání hesla (rovněž je třeba znovu soubor načíst, jako v předchozím případě), jinak nemůže extrakce dat proběhnout.

Po úspěšném načtení souboru se skrytými daty (obr. A.6) je možné si přehrát zvukový záznam ze souboru (podobně jako v přehrávači v hlavním okně). Dále jsou uživateli zobrazeny tyto informace o načteném souboru (steganogramu):

- délka záznamu (ve tvaru *mm:ss*),
- počet bitů na vzorek záznamu,
- počet kanálů záznamu,
- vzorkovací kmitočet záznamu v kHz,
- způsob rozprostření dat,
- hloubka modifikace (počet přepsaných LSB na vzorek),
- množství skrytých dat v bajtech,
- příznak komprese dat.

Pokud proběhly výše uvedené kontroly v pořádku, příp. bylo zadáno heslo a provedeno znovunačtení steganogramu, je možné provést extrakci dat do vybraného souboru (objeví se dialog pro výběr cesty a zadání jména souboru) tlačítkem *Extrahovat*.

A.2.6 Konfigurační soubor

Některé vlastnosti a parametry programu jsou nastavitelné v konfiguračním souboru (`config.ini`). Konfigurační parametry jsou rozděleny podle příslušnosti do sekcí `general` (obecná nastavení), `plot` (nastavení týkající se grafické reprezentace zvukových signálů) a `hexview` (nastavení prohlížeče dat).

Strukturou se jedná o klasický INI soubor – sekce jsou označeny hranatými závorkami, řádky s komentáři začínají středníkem a nastavení jednotlivých parametrů je zadáno v podobě `parametr=hodnota`. Jednotlivé parametry jsou ve výchozím souboru stručně okomentovány.

Sekce general

Tab. A.1: Konfigurační parametry v sekci general

Parametr	Význam/popis
`tmpfilename`	Jméno, resp. cesta k souboru pro dočasné uložení steganogramu.
`log_show`	Volba, zda zobrazovat textové pole s protokolem činností v dolní části okna (1), nebo pole nezobrazovat (0)
`log_bgcolor`	Barva pozadí pole s protokolem činností (v „HTML“ formátu `#RRGGBB`)

Sekce plot

Nastavení v této sekci (dle tab. A.2.6) se týkají vykreslování grafů.

Sekce hexview

Tato nastavení (dle tab. A.3) se týkají Hex prohlížeče. Hodnoty veškerých barev jsou v „HTML“ tvaru – `#RRGGBB`.

A.3 Poznámky k implementaci

Program je napsán v jazyce Python 3 (testováno s verzí 3.3.2) a dále využívá tyto knihovny/moduly, které nejsou součástí výchozí instalace jazyka Python:

- `pyaudio` pro přehrávání zvukových záznamů,
- `pycrypto` pro šifrování blokovou šifrou AES-256,
- `matplotlib` pro vykreslování grafů (průběhu) zvuk. signálů; modul `matplotlib` dále závisí na modulech `python-dateutil`, `six`, `pyparsing` a `numpy` (jejich verzi pro MS Windows je možné stáhnout z `http://www.lfd.uci.edu/~gohlke/pythonlibs/`)

Samostatně spustitelná verze programu s přibalenými knihovnami byla vytvořena pomocí nástroje `cx_freeze`. Program byl vytvořen a otestován pod OS Microsoft Windows 7, ale mělo by být možné jej v případě potřeby bez modifikací spustit i pod jinými OS s nainstalovaným interpretem jazyka Python a potřebnými moduly.

Tab. A.2: Konfigurační parametry v sekci plot

Parametr	Význam/popis
`color_new`	Barva pro vykreslení průběhu nového zvukového signálu – se skrytými daty (ve tvaru `#RRGGBB`)
`color_orig`	Barva pro vykreslení průběhu původního zvukového signálu (ve tvaru `#RRGGBB`)
`alpha_new`	Průhlednost čáry při vykreslení průběhu nového zvuk. signálu (hodnota = číslo s desetinnou tečkou od 0 do 1)
`alpha_orig`	Průhlednost čáry při vykreslení průběhu původního zvuk. signálu (hodnota = číslo s desetinnou tečkou od 0 do 1)
`grid`	Zobrazení mřížky v grafu (`0` – nezobrazovat, `1` – zobrazit)
`sampskip`	Koeficient podvzorkování (vzorků původního i nového signálu) při vykreslování grafů (podvzorkování 1:n). Pokud je zadána hodnota `-1`, provede se podvzorkování automaticky na základě zadaného max. počtu vzorků.
`axistype`	Typ hodnot na ose x: `0` = hodnoty na ose x představují pořadí vzorku [-], `1` = hodnoty na ose x představují čas [s].
`max_samples`	Maximální počet vzorků (vykreslovaných hodnot), které budou načteny a vykresleny do grafu. Pokud je nastaveno automatické podvzorkování, tato hodnota určuje spolu s počtem vzorků zvukového signálu koeficient podvzorkování.

Tab. A.3: Konfigurační parametry v sekci hexview

Parametr	Význam/popis
`fgcolor`	barva textu
`bgcolor`	barva pozadí
`fontfamily`	rodina fontu
`fontsize`	velikost fontu
`selfgcolor`	barva manuálně označeného textu
`selbgcolor`	barva pozadí manuálně označeného textu
`diff_fgcolor`	barva textu (bajtů), který se liší v původním a novém souboru WAV
`diff_bgcolor`	barva pozadí textu (bajtů), který se liší v původním a novém souboru WAV
`dataposbgcolor`	barva pozadí ve sloupci s označením pozic zobrazených řádků v rámci souboru
`dataposdisplay`	zobrazení sloupce s označením pozic zobrazených řádků v rámci souboru (0 = nezobrazovat, 1 = zobrazit).
`wrap`	počet znaků (bajtů) souboru, které budou zobrazeny na jednom řádku

B TEXT LABORATORNÍ ÚLOHY

V této příloze je uvedeno zadání laboratorní úlohy a text poznámek pro vedoucí cvičení, přičemž texty jsou po stránce formátování upravené pro začlenění do této práce. Zároveň jsou dostupné ve formátu LibreOffice (OpenDocument) pro usnadnění případných úprav a jsou součástí obsahu přiloženého disku CD.

B.1 Teoretický úvod

B.1.1 Steganografie

Steganografie je oborem, který se zabývá ukrýváním informací (netriviálním způsobem). Ukrývání informací má velmi dlouhou historii, avšak steganografie jako samostatný obor byla vyčleněna až ve 20. století, kdy se jí zejména s rozvojem informačních technologií dostalo větší pozornosti než dříve. V informačních technologiích je pro skrývání informací (dat) možné využít celou řadu *nosičů* (tj. entit, do kterých jsou skrývány informace), především se nabízí obrazové nebo zvukové soubory. Po skrytí tajných dat (tajné zprávy) do nosiče vzniká *steganogram*.

Pro skrývání dat do zvukových záznamů existuje více technik, založených především na základě znalosti psychoakustického modelu (tj. vnímání zvuku člověkem). Tyto techniky mohou využívat například maskování v kmitočtové nebo časové oblasti nebo neschopnost lidského sluchu poznat do určité míry zvýšenou hladinu vneseného šumu, čehož využívá i technika označovaná jako *LSB* (Least Significant Bits, nejméně významné bity). Při využití této techniky se skrývá vždy n bitů tajné zprávy do n LSB, což lze u zvukových záznamů snadno realizovat, protože každý ze vzorků záznamu je reprezentován jako několikabitové (typicky 8bitové nebo 16bitové) číslo. S rostoucím počtem LSB nahrazených bity tajné zprávy roste i míra poškození zvukového signálu, a proto je třeba najít vhodnou mez, při které je možné do záznamu vložit co největší množství informace, ale zároveň nepoškodit záznam natolik, aby byla změna snadno postřehnutelná. Množství informace, které lze do záznamu vložit, se nazývá steganografická kapacita (C) a závisí na počtu bitů na vzorek n_b, počtu zvukových kanálů n_k, vzorkovacím kmitočtu f_{vz} a délce trvání záznamu t:

$$C = n_b \cdot n_k \cdot f_{vz} \cdot t \text{ [bit]} \tag{B.1}$$

B.1.2 Formát WAV

Formát *WAV* je aplikací formátu *RIFF* (Resource Interchange File Format), přičemž ukládaná data nebo metadata jsou členěna do bloků (resp. podbloků). Každý blok

začíná čtyřbajtovým identifikátorem (blok nejvyšší úrovně u formátu WAV je označen jako `RIFF`), za ním následují 4 B (little endian, tedy na prvním místě nejméně významný bajt) specifikující délku bloku. Následuje podblok `WAVE`, dále rozdělený na podblok `fmt` (obsahuje informace o typu vzorků, počtu kanálů, vzorkovacím kmitočtu, počtu přehraných bajtů za sekundu, velikosti vzorku v bajtech a počtu bitů na vzorek. Vlastní zvuková data jsou uložena v dalším podbloku `data` (za identifikátorem bloku `data` opět následují 4 B označující délku bloku, a dále jsou již uloženy vzorky). V případě 8bitových vzorků je každý vzorek uložen jako jeden bajt a reprezentuje kladné celé číslo (tedy celé číslo z rozsahu $[0; 255]$). 16bitové vzorky jsou ukládány do dvou bajtů v pořadí little endian a představují celé číslo se znaménkem, tj. celé číslo z rozsahu $[-32768; 32767]$.

B.1.3 Program Stego

Charakteristika

Program Stego slouží ke skrývání dat libovolného charakteru do zvukových souborů typu WAV za pomoci techniky LSB. Podporuje nekomprimované soubory s 8 a 16 bity na vzorek a jedním nebo dvěma zvukovými kanály (mono/stereo). Umožňuje uživateli zvolit si hloubku skrývání dat (tedy počet LSB, které budou ve vzorku nahrazeny adekvátně dlouhou sekvencí bitů ze skrývaných dat) a způsob rozprostření dat ve zvukovém záznamu:

- **Sekvenční rozprostření** (výchozí volba) – vzorky pro skrytí dat jsou vybírány sekvenčně od začátku dostupných vzorků zvukového záznamu.
- **Rovnoměrné rozprostření** – vzorky pro skrytí dat jsou vybírány rovnoměrně přes celou posloupnost dostupných vzorků záznamu.
- **Pseudonáhodné rozprostření** – pozice vzorků pro skrytí dat je vybírána v rámci dostupných vzorků iterativně pseudonáhodným způsobem za využití hašovací funkce SHA2-256. Pro tento způsob je nutné zadat heslo, na jehož základě je vypočítána první pozice upravovaných vzorků v záznamu, a dále iterativně i další pozice.

Před vlastními skrývanými daty je do zvukového záznamu na jeho počátku vloženo ještě záhlaví s informacemi o skrývaných datech (počet přepisovaných LSB, délka uložených dat, příznak komprese dat, haš dat sloužící jako kontrolní součet). Tyto informace jsou do vzorků ukládány s hloubkou modifikace 1 LSB na vzorek. Dostupné vzorky pro ukládání skrývaných dat tedy následují až za vzorky, ve kterých je uloženo záhlaví. Program umožňuje volitelnou kompresi skrývaných dat (což v některých případech dovoluje uložit do záznamu stejné množství dat s menším poškozením zvukového záznamu, resp. při stejném poškození uložit větší množství dat). Skrývaná data

je možné zašifrovat (šifrování se provádí včetně záhlaví) blokovou šifrou *AES-256*. Mimo skrývání dat samozřejmě program disponuje i funkcí pro extrakci skrytých dat z WAV souboru (steganogramu). Původní zvukový záznam i nově vytvořený záznam (steganogram) jsou reprezentovány ve formě grafu průběhu zvukového signálu v čase (resp. v rámci sekvence vzorků). Další možnost reprezentace je zobrazení původního a nového souboru jako binárních dat po bajtech.

Ovládání programu

Hlavní okno programu (obr. A.2) sestává z těchto hlavních celků:

1. **Grafická reprezentace** průběhu zvukového signálu (hodnot vzorků v sekvenci vzorků, resp. v čase). Zobrazenou oblast je možné přibližovat, oddalovat a posunovat (blíže viz následující bod). Pro lepší orientaci je ve výchozím stavu zobrazena mřížka, je možné ji vypnout přepínačem na pravé straně okna pod oblastí grafu.
2. **Tlačítka pro úpravu zobrazení grafu**. Prvním tlačítkem se přepíná na funkci posunu/změny měřítka. Při pohybu myší v oblasti grafu se stiskutým levým tlačítkem dochází k posunu zobrazené části signálu. Při pohybu se stisknutým pravým tlačítkem se mění měřítko zobrazení. Druhým tlačítkem se zapíná funkce přibližování/oddalování vybrané obdélníkové oblasti (přibližování kliknutím a tažením levým tlačítkem myši, oddalování tlačítkem pravým).
3. **Nastavení hloubky modifikace** (počtu LSB), typu rozprostření, aktivace komprese dat nebo šifrování. Pro šifrování a/nebo volbu pseudonáhodného rozprostření je nutné zadat heslo (po kliknutí na tlačítko *Nastavit heslo*). Za předpokladu, že byl načten vstupní zvukový soubor (nosič) a data určená pro skrytí, a steganografická kapacita nosiče je pro uložení dat dostatečná, je možné provést skrytí dat tlačítkem *Skrýt data*. Poté, co proběhlo skrytí dat, je možné zobrazit porovnání vstupního a výstupního souboru ve formě posloupnosti bajtů a jejich hexadecimální reprezentace (obr. A.3) kliknutím na tlačítko *HEX zobrazení*.
4. **Informace**. V této oblasti se zobrazují informace o vstupním zvukovém souboru, a dále množství skrývaných dat v nekomprimované formě.
5. **Přehrávač** původního a nově vytvořeného zvukového záznamu. Umožňuje přehrát a zastavit nebo pozastavit zvukové záznamy a na základě sluchového vjemu porovnat, jaký vliv mají skrytá data na kvalitu vzniklého záznamu se skrytými daty.

6. **Textové pole se záznamem provedených akcí** (nemusí být zobrazeno – záleží na nastavení v konfiguračním souboru). Podává přehled o tom, jaké akce (načtení souborů, provedení skrytí dat aj.) byly provedeny, a v jakém čase.

Vlastní načtení nosiče se provede v nabídce *Akce-Načíst vstupní WAV soubor*, podobně načtení skrývaných dat v nabídce *Akce-Načíst soubor s daty pro skrytí*. Steganogram je možné uložit příkazem z nabídky *Akce-Uložit výstupní WAV soubor (steganogram)*. Extrahovat skrytá data z vytvořeného steganogramu lze volbou *Akce-Extrakce skrytých dat z WAV souboru*. V případě, že jsou data šifrovaná nebo byla použita pseudonáhodná metoda rozprostření, je zapotřebí zadat na vyzvání heslo, jinak nemůže extrakce dat proběhnout!

Při načítání souborů se provádí kontrola, zda je steganografická kapacita nosiče dostatatečná. Pokud by docházelo k problémům z důvodu této kontroly při již načtených datech, je možné program uvést do výchozího stavu příkazem z nabídky *Akce-Reset vstupních dat*. V případě, že potřebujete podrobnější informace k používání programu, využijte Uživatelskou příručku, která obsahuje detailní popis programu z uživatelského hlediska (volba *Nápověda-Zobrazit nápovědu*).

Skrývání dat do WAV souborů a vliv na kvalitu záznamu

Pro ověření vlivu různého množství dat (s různou hloubkou modifikace) na různé typy zvukových záznamů jsou k dispozici soubory s daty pro skrytí – JPEG fotografie `selma*.jpg` a text *Lorem ipsum* `lorem_ipsum*.txt` (o různých velikostech) a dále sada zvukových souborů WAV s různým charakterem záznamu (hudba, mluvené slovo, bílý šum, aj.) ve verzích s 16 b/vzorek a 8 b/vzorek. Jejich podrobný seznam a popis je uveden v tab. B.1.4.

a) **Nejprve se blíže seznamte se způsoby rozprostření dat do vzorků zvukového záznamu:**
   1. Načtěte zvukový soubor `03_ticho_15s.wav`, který obsahuje 15 s ticha (vzorků s hodnotou 0).
   2. Načtěte soubor `lorem_ipsum_20K.txt` (obsahuje přibližně 20 kB pseudolatinského textu Lorem ipsum).
   3. Ponechte vybraný sekvenční způsob skrytí, nastavte hloubku modifikace na 16 LSB/vzorek a proveďte skrytí. Prozkoumejte vzniklý zvukový záznam v grafické podobě a porovnejte jej s předchozím záznamem. Přehrajte si původní záznam a nově vzniklý záznam.
   4. Stejně jako v bodu 3 proveďte skrytí, ale tentokrát vyberte rovnoměrné rozprostření, a porovnejte grafickou reprezentaci původního a nového záznamu a přehrajte si nový záznam.

5. Dále proveďte totéž s pseudonáhodným způsobem rozprostření (je nutné zadat heslo!).
6. Jak se projevují jednotlivé způsoby rozprostření? Jak se vzájemně liší?
7. Načtěte soubor `lorem_ipsum_50K.txt` a proveďte rozprostření sekvenčním způsobem pro nastavený počet přepisovaných LSB 1 až 16. Pozorujte, při jaké hodnotě začne být ovlivnění původního signálu slyšitelné.

b) **Dále vyzkoušejte, jakým způsobem se projeví skrytí dat do bílého šumu:**

1. Načtěte zvukový soubor `05_bily_sum_30s.wav` a soubor s daty `lorem_ipsum_500K.txt`.
2. Nastavte hloubku modifikace na 16 LSB na vzorek, proveďte skrytí dat, prozkoumejte grafickou reprezetaci zvukového záznamu a poslech modifikovaného záznamu, a to pro všechny způsoby rozprostření. Jak se vzájemně liší? V čem spočívá nevýhoda při skrývání textových dat při nastavení hloubky modifikace 16, příp. 8 LSB/vzorek (prozkoumejte původní a nově vytvořený soubor v HEX prohlížeči).
3. Aktivujte šifrování a opět proveďte skrytí dat pro stejnou hloubku modifikace (metoda rozprostření může být libovolná). V čem se liší výsledek od výsledků z předchozího bodu?
4. Deaktivujte šifrování, aktivujte kompresi a skryjte data sekvenční metodou. K jaké podstatné změně došlo (zaměřte se na úplný začátek záznamu)? V jakých případech by k takové změně nedošlo?
5. Načtěte soubor s daty `selma_500K.jpg` a skryjte jej s hloubkou modifikace 16 LSB/vzorek (s libovolnou metodou rozprostření). Porovnejte s bodem 2 a 3.

c) **Vyzkoušejte si rovněž skrývání dat do různých typů zvukových záznamů:**

1. Do souboru `10_wall.wav` skryjte soubor `selma_500K.jpg` při hloubce modifikace 10 LSB/vzorek sekvenční metodou rozprostření a poslechněte si výsledný záznam. Poté proveďte skrytí opakovaně, ale při rovnoměrném rozprostření. Jak se výsledné záznamy liší?
2. Do některého ze souborů s hudbou (např. `10_wall.wav`) skryjte soubor `selma_500K.jpg` pomocí sekvenční metody rozprostření a pokuste se zjistit, při jaké hloubce modifikace ještě změna není sluchem rozpoznatelná. Porovnejte, jak se vliv na zvukovou nahrávku liší v tichých a dynamických pasážích.
3. Opakujte bod 2 pro některý ze souborů, ve kterých jsou použity 8bitové vzorky (`*_8b.wav`) a porovnejte se situací, kdy jsou použity 16bitové vzorky.

4. Na základě poznatků z předchozích bodů si vyzkoušejte skrytí dat do různých dalších typů zvukového záznamu (zvolte několik hodnot hloubky modifikace a příp. způsobů rozprostření dle vlastního uvážení). Zaměřte se především (ale nikoliv výlučně) na zvukové soubory `04_sidliste_60s_8kHz.wav` a `06_interview.wav`. Můžete si také vyzkoušet uložení záznamu se skrytými daty a opětovnou extrakci dat.

B.1.4 Použité zvukové záznamy

Pro úlohu jsou k dispozici zvukové soubory uvedené v následující tabulce. Většina z nich jsou záznamy uvolněné pod svobodnou licencí (*Creative Commons* ve verzi kompatibilní s tímto způsobem použití), příp. se jedná o záznamy vytvořené přímo pro tuto úlohu. Některé ze záznamů byly pro použití v úloze upraveny (zkráceny, příp. z nich byly odstraněny části obsahující ticho aj.).

Tab. B.1: Zvukové záznamy použité v laboratorní úloze

Soubor	Popis	Autor
`01_city_centre.wav`	ruch v centru města (*City Centre Sound*)	Hopeinwave
`02_paper_rip.wav`	zvuk trhání papíru (*Slow Paper Rip Sound*)	Mike Koenig
`03_ticho_15s.wav`	prázdný zvukový záznam, 15 s	Jiří Kortus
`04_sidliste_60s_8kHz.wav`	záznam ruchu klidné části sídliště na kraji města v nepracovní den (pořízeno mobilním telefonem a zesíleno – obsahuje šum)	Jiří Kortus
`05_bily_sum_30s.wav`	bílý šum s rozkmitem od min. do max. možné hodnoty vzorků	Jiří Kortus
`06_interview.wav`	interview - *Talking About Music Videos*	Marissa Nadler
`07_thunder_sound.wav`	zvuk hromů (*Thunder FX Sound*)	Grant Evans
`08_na_patou.wav`	skladba *Na pátou*	Gem Reflection
`09_laugh_tracks.wav`	smích publika (*Laugh Tracks, Invisible Crowd, Interjections*)	Jefferson Kielwagen
`10_wall.wav`	skladba *And Now We Stand Against The Wall*	Garmisch
`11_intelligent.wav`	skladba *They Seem To Be Intelligent*	Garmisch

Text *Lorem ipsum* byl získán z `http://en.wikisource.org/wiki/Lorem_ipsum`, zvukové záznamy pochází z následujících zdrojů:

[01] `http://soundbible.com/2000-City-Centre.html`

[02] `http://soundbible.com/1925-Slow-Paper-Rip.html`

[06] `http://freemusicarchive.org/music/Marissa_Nadler/Live_at_WFMU_on_Dark_Night_of_the_Soul_with_Julie_on_August_23_2011_1054/Marissa_Nadler_-_08_-_Talking_About_Music_Videos`

[07] `http://soundbible.com/2053-Thunder-Sound-FX.html`

[08] `http://www.jamendo.com/en/track/1055386/na-patou`

[09] `http://freemusicarchive.org/music/Jefferson_Kielwagen/Radius_PATCH_05_Absence/RAD0043`

[10, 11] `http://freemusicarchive.org/music/Garmisch/Fishes/`

B.2 Poznámky pro vedoucí cvičení

B.2.1 Obecné poznámky

Je důležité, aby si studenti při zkoumání změn ve zvukovém záznamu uvědomili jejich charakter. Především, kde v rámci záznamu by se změny měly projevit (dle způsobu rozprostření), jak dlouhá je ovlivněná část záznamu od jeho začátku (u sekvenčního typu rozprostření) a jak velké budou změny hodnoty vzorku. Změny např. pro hloubku modifikace 1 LSB na vzorek představují změnu hodnoty maximálně o 1 (takže je třeba v zobrazeném grafu provést velké přiblížení), avšak rostou exponenciálně s hloubkou modifikace. To by si též studenti měli uvědomit i v úkolech, ve kterých zkoumají, kdy začnou být změny způsobené skrytými daty slyšitelné (to rovněž záleží i na dalších faktorech, jako soustředěnost posluchače, fyzické dispozice, kvalita reprodukční soustavy apod.).

B.2.2 Část a) – seznámení se způsoby rozprostření

Po skrytí souboru `lorem_ipsum_20K.txt` do zvukového záznamu `03_ticho_15s-.wav` s hloubkou modifikace 16 LSB dojde při sekvenčním typu rozprostření ve zvukovém záznamu k vytvoření úseku šumu s dobou trvání přibližně 0,2 s.

Při rovnoměrném rozprostření bude charakter záznamu zcela odlišný (nebude se již jednat o šum) z důvodu vytvoření vzorků s nenulovou hodnotou rovnoměrně v celé posloupnosti vzorků (a mezi nimi budou rovnoměrné mezery).

Při pseudonáhodném rozprostření dojde k rozložení skrývaných dat přibližně přes celou délku zvukového záznamu, ale v nepravidelném pořadí a s nepravidelnými rozestupy (s ponechanými původními vzorky). Výsledný zvukový záznam bude mít charakter šumu s výrazným praskáním.

Slyšitelné ovlivnění původního záznamu (tj. vznik zvukového signálu, který bude možné zaregistrovat sluchem) závisí na tom, jak je nastavena výstupní hlasitost (resp. zesílení), na kvalitě reprodukčního zařízení, úrovni okolního hluku a citlivosti sluchu a psychickém rozpoložení posluchače. Při dostatečném zesílení může být slyšitelná i změna, která nastala v důsledku skrytí dat s modifikační hloubkou 1 LSB na vzorek pro 16bitové vzorky. To však platí prakticky jen v tomto případě, kdy jsou data skrývána do vzorků s nulovou hodnotou. Při skrývání do skutečného zvukového záznamu (hlas, hudba, apod.) by posluchač tak malou změnu neregistroval, resp. hlasitost (z důvodu potřebného zesílení) by byla tak velká, že by posluchač ani záznam nemohl poslouchat. Pokud by však zůstala nastavena hlasitost na stejné úrovni jako v předchozích případech, nacházel by se hledaný práh přibližně na hodnotě hloubky modifikace 6-8 LSB/vzorek.

B.2.3 Část b) – skrývání dat do zvukového záznamu s bílým šumem

Při sekvenčním způsobu rozprostření souboru s daty `lorem_ipsum_500K.txt` do záznamu `05_bily_sum_30s.wav` se první cca 2,5 s projevuje vznikem nového zvukového signálu s opakujícím se charakterem, který neodpovídá bílému šumu. Dále následuje pouze původní signál, tedy bílý šum.

Při rovnoměrném rozprostření se při poslechu charakter záznamu více blíží bílému šumu, ale mimo něj obsahuje i další tón, který v bílém šumu obsažen nebyl.

Volba pseudonáhodného rozprostření neovlivní původní vzorky rovnoměrně, ale na pseudonáhodných místech, kdy navíc n-tice LSB obsahující bity skrývaných dat nejsou ve vzorcích obsaženy v původním pořadí. Z hlediska poslechového vjemu jsou oba zvukové záznamy totožné.

Při skrytí dat, která byla předtím zašifrována, dostanou data podobu pseudonáhodné posloupnosti, což se po skrytí projeví vznikem signálu s charakterem bílého šumu (podobně jako u pseudonáhodného způsobu rozprostření).

Je-li aktivována komprese, dojde k razantnímu snížení objemu skrývaných dat, takže záznam je ovlivněn jen přibližně do 0,04 s od jeho začátku. Zda k podobnému jevu dojde, záleží především na typu skrývaných dat (např. u textu se dá předpokládat podstatné snížení objemu skrývaných dat, na rozdíl od dat typu

obrázek JPEG nebo zvukový záznam formátu MP3, které již neposkytují příliš prostoru pro obecnou bezeztrátovou kompresi z důvodu nízké redundance).

Po skrytí obrázku JPEG do zvukového záznamu bude mít výsledný záznam opět charakter bílého šumu, vzhledem k tomu, že obsah souboru se blíží pseudonáhodné sekvenci bajtů.

B.2.4 Část c) – skrývání dat do různých typů zvukových záznamů

Po skrytí souboru `selma_500K.jpg` do souboru `10_wall.wav` při hloubce modifikace 10 LSB/vzorek sekvenční metodou dojde k podstatnému ovlivnění záznamu na jeho počátku, ale jen poměrně krátce, avšak zcela zřetelně (první 4 s), dále je už záznam neovlivněný. V případě rovnoměrného rozprostření je ovlivněn záznam v celé délce, ale v podstatně menší míře, kdy v hlasitějších nebo dynamických částech záznamu nemusí být ovlivnění zřetelné (dle okolností poslechu). V tomto případě je situace podobná jako u úkolu a) (takže nelze stanovit konkrétní univerzálně platnou hodnotu hloubky modifikace, pro kterou by změny nebyly postřehnutelné) a opět zde platí, že ovlivnění (poškození) záznamu je patrné v tišších pasážích.

Při použití záznamů s 8bitovými vzorky je při stejné hloubce modifikace míra poškození podstatně vyšší. Při použití 8b vzorků může vzorek nabývat celkem 256 různých hodnot, zatímco při použití vzorků 16bitových může vzorek nabývat 65536 hodnot, což je druhá mocnina prvního případu. Např. poškození záznamu při hloubce modifikace 1 LSB/vzorek u záznamu s 8b vzorky přibližně odpovídá hloubka modifikace 8 LSB/vzorek při použití záznamu s 16b vzorky. Je zřejmé, že jak z hlediska steganografické kapacity, tak i míry poškození záznamu skrývanými daty, jsou výhodnější záznamy s 16bitovými vzorky.

Pro skrývání dat jsou výhodnější záznamy obsahující minimální množství tichých úseků (to může být problematické např. pro záznamy s rozhovory nebo mluveným slovem). Rovněž jsou vhodné záznamy, které přirozeně obsahují šum, např. nahrávky z nepříliš kvalitního záznamového zařízení (v našem případě jde o nahrávku pořízenou mobilním telefonem a zesílenou). U takové nahrávky lze šum bez většího podezření považovat za její přirozenou součást, takže je vhodná pro skrytí tajných dat.

OBSAH PŘILOŽENÉHO CD

Součástí této práce jsou následující přílohy v elektronické podobě, které jsou uloženy na přiloženém disku CD:

- Text diplomové práce ve formátu PDF
- Zadání laboratorní úlohy ve formátu OpenDocument (LibreOffice)
- Poznámky pro vedoucí cvičení ve formátu OpenDocument (LibreOffice)
- Zdrojové kódy programu Stego včetně uživatelské příručky
- Program Stego pro MS Windows XP/7 ve spustitelné podobě včetně potřebných knihoven (32b a 64b verze)