



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

DEPARTMENT OF COMPUTER SYSTEMS

## **DIGITÁLNÍ STEGANOGRAFIE PRO SPUSTITELNÉ SOUBORY**

DIGITAL STEGANOGRAPHY FOR EXECUTABLES

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**L'UBOŠ BEVER**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. JOSEF STRNADEL, Ph.D.**

**BRNO 2022**

## Zadání bakalářské práce



24329

Student: **Bever Luboš**

Program: Informační technologie

Název: **Digitální steganografie pro spustitelné soubory**  
**Digital Steganography for Executables**

Kategorie: Bezpečnost

Zadání:

1. Vytvořte přehled metod z oblasti digitální steganografie, zúženěji a detailněji pak přehled z podoblasti steganografie pro skrývání informace ve spustitelných souborech (dále jen "steganografie"), jejich vlastností a shrňte současný stav a trendy v této zúžené podoblasti.
2. Zvolte typ skrývané informace (textová, obrazová apod.) a její vlastnosti. Na základě existující či vlastní analýzy formátů spustitelných souborů a typu skrývané informace zvolte vhodné formáty spustitelných souborů a vhodné steganografické metody.
3. V souladu s bodem 2 připravte vhodnou sadu dat pro ověřování vlastností zvolených steganografických metod a navrhnete mechanismus vyhodnocování jejich vlastností na základě těchto dat.
4. Implementujte několik existujících metod steganografie pro spustitelné soubory, zvažte jejich modifikace, popř. návrh a implementaci vlastních metod.
5. Vhodně ověřte funkčnost implementovaných metod; vyhodnoťte jejich vlastnosti a porovnejte je jak navzájem, tak s daty z několika publikací.
6. Shrňte dosažené výsledky, diskutujte možné směry využití a rozvoje předloženého řešení.

Literatura:

- Dle pokynů vedoucího.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Strnadel Josef, Ing., Ph.D.**

Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 29. října 2021

## Abstrakt

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém (slovenském) jazyce.

[[Uvedte kontext důležitý pro pochopení problematiky, cíle práce, hypotézy, zvolené metody a postupy řešení.]] [[POZRIET SABLONU]]

## Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

## Klíčové slová

Sem budou zapsána jednotlivá klíčová slova v českém (slovenském) jazyce, oddělená čárkami.

[[Jedna z nejlepších rad pro psaní článků (a obecně odborného textu), kterou jsem kdy slyšel, není úplně intuitivní a samozřejmá. Napište si klíčová slova, jež by člověk měl napsat do vyhledávače, aby mu vypadlo Vaše dílo jako relevantní odpověď. Popuštěte uzdu fantazii, klidně to vezměte ze široka. Přemýšlejte o aplikacích Vaší práce. O souvislostech.

Sepište všechna klíčová slova, bude to na několik řádků. Klíčové slovo je i sousloví – typicky dvou nebo tří slov. Vyberte z nich ta důležitá. K tomu je potřeba intuice a zkušenost. Kde ty vzít, nevím, ale vždycky se můžete s někým (např. vedoucím práce) poradit. Až budete psát svůj třicátý odborný text, půjde to celkem hladce. Všechna důležitá klíčová slova se musí objevit v nadpisu článku nebo v nadpisech kapitol.]]

## Keywords

Sem budou zapsána jednotlivá klíčová slova v anglickém jazyce, oddělená čárkami.

## Citácia

BEVER, Ľuboš. *Digitálna steganografia pro spustiteľné súbory*. Brno, 2022. Bakalárska práca. Vysoké učenie technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Josef Strnadel, Ph.D.

# Digitální steganografie pro spustitelné soubory

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Josefa Strnadela, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....

Luboš Bever  
22. apríla 2022

## Podakovanie

Chcem sa poďakovať svojmu vedúcemu práce Ing. Josefovi Strnadelovi, Ph.D. za odborné rady, cenné pripomienky, trpezlivosť, ochotu a pomoc, ktorú mi poskytol pri písaní bakalárskej práce. Moja vďaka tiež patrí rodine a priateľke za podporu nielen pri písaní tejto práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Skrývanie dát v rámci iných dát</b>	<b>4</b>
2.1	Úvod do digitálnej steganografie . . . . .	4
2.2	Skrývanie dát v texte . . . . .	11
2.3	Skrývanie dát v obrázkoch . . . . .	12
2.4	Skrývanie dát vo zvuku . . . . .	12
2.5	Skrývanie dát vo videu . . . . .	13
2.6	Skrývanie dát v internetovej sieti . . . . .	14
2.7	Iné možnosti ukrytia dát . . . . .	14
<b>3</b>	<b>Digitálna steganografia spustiteľných súborov a ich analýza</b>	<b>15</b>
3.1	Analýza formátu ELF . . . . .	15
3.2	Analýza formátu PE . . . . .	19
3.3	Spôsoby ukrytia dát v spustiteľných súboroch . . . . .	24
3.4	Existujúce aplikačné nástroje . . . . .	25
<b>4</b>	<b>Návrh ...</b>	<b>28</b>
4.1	Architektúra aplikácie . . . . .	28
4.2	Popis zvolenej steganografickej metódy . . . . .	28
4.3	Popis skúmania vlastnej modifikácie substitučných tried . . . . .	28
4.4	Mechanizmus vyhodnocovania vlastností použitých steganografických metód . . . . .	28
<b>5</b>	<b>Implementácia ...</b>	<b>29</b>
<b>6</b>	<b>Testovanie a experimenty ...</b>	<b>30</b>
<b>7</b>	<b>Záver</b>	<b>31</b>
	<b>Literatúra</b>	<b>33</b>
<b>A</b>	<b>Špecifikácia metód jednotlivých digitálnych steganografií</b>	<b>38</b>
A.1	Textová steganografia . . . . .	38
A.2	Obrazová steganografia . . . . .	41
A.3	Zvuková steganografia . . . . .	44
A.4	Videosteganografia . . . . .	47

# Kapitola 1

## Úvod

**[U]**kladanie, prijímanie a odosielanie súkromných informácií je základnou aktivitou všetkých používateľov internetu. Avšak, informačná bezpečnosť je vo vysokom záujme najmä vládnych organizácií. Steganografia, alebo skrývanie informácií, predstavuje významnú hrozbu pre rôzne inštitúcie či spoločnosti a ich digitálne produkty. Existujú záznamy, že bola tiež použitá pri plánovaní teroristického útoku na Svetové obchodné centrum v New Yorku z roku 2001.

**[N]**eoprávnený prístup k dôverným informáciám je v súčasnom digitálnom svete veľkým lákadlom pre útočníkov. Existujú tak dôvody, kvôli ktorým vznikajú rôzne snahy a opatrenia proti takýmto činom. Je preto zrejmé, že koncept steganografie majú v záujme rozvíjať obe strany tejto situácie. Keďže veľkosť multimediálnych objektov je pre ľudské porozumenie enormná, bolo nájdených veľa spôsobov, ako steganografiu pomerne bezpečne použiť. Skutočnosťou však zostáva, že s príchodom nových metód prichádzajú aj techniky, ktoré ich bezpečnosť ochromujú. Ide o nekonečný cyklus podobný kryptografii a kryptoanalýze.

**[S]**krývanie dát v rámci spustiteľných súborov tvorí osobitnú kapitolu celej problematiky steganografie. Ide o podobnú praktiku vkladania dát do programov, ako je tomu pri injekcii škodlivých častí kódu. Nielenže zámena jediného bitu informácie môže znefunkčniť celý program, ale je nutné myslieť aj na detekciu tajných dát antivírusovými programami. Tie sú dnes schopné detegovať rôzne, dokonca aj predtým nimi nepoznané, anomálie. Všetky tieto fakty veľmi znevýhodňujú skrývanie v rámci spustiteľných súborov, a preto doposiaľ v tomto uplynulý výskum je v porovnaní s ostatnými druhmi steganografie, neporovnateľne menší.

**[E]**xistuje pomerne malé množstvo techník, ktorými je možné bezpečne ukryť dáta v programoch. Navyše, tieto techniky sú významne závislé od inštrukčnej sady procesorov a sú založené na hlboknej analýze formátov spustiteľných súborov.

**[E]**xperimentálne zhodnotenie bezpečnosti a použiteľnosti techník je dôležitým konceptom pri návrhu nových metód akejkoľvek steganografie. Cieľom tejto práce je implementácia niektorých existujúcich steganografických metód pre spustiteľné súbory a návrh mechanizmu, ktorý by bol schopný tieto metódy porovnať a zhodnotiť ich vlastnosti.

**[N]**a začiatok, kapitola 2 poskytuje čitateľovi komplexný prehľad na digitálnu steganografiu, ktorý si je možné ešte rozšíriť prílohou A. Obsahom kapitoly 3 je analýza formátov spustiteľných súborov, s ktorými táto práca počíta, predstavenie súčasných metód steganografie spustiteľných súborov a kapitola končí prehľadom aktuálne dostupných aplikačných nástrojov v tejto oblasti. Cieľom kapitoly 4 je návrh vytvorenej aplikácie, týkajúci sa najmä použitých metód, a idea vyhodnotenia metrík, ktorým sa budú jednotlivé metódy testovať

a porovnávať. Plynule nadväzuje kapitola 5, ktorá popisuje implementáciu navrhnutej aplikácie. Dôležitým prvkom tejto práce je popis uskutočneného testovania a experimentovania s implementovanou aplikáciou. Na záver sa v kapitole 7 zhodnotí dosiahnutý cieľ, získané výsledky a navrhnu sa možné vylepšenia.

**[[Improving embedding efficiency for digital steganography by exploiting similarities between secret and cover images – DO UVODU**

Rýchly pokrok v digitálnych komunikačných technológiách a obrovský nárast počítačového výkonu spôsobili exponenciálny nárast používania internetu na rôzne komerčné, vládne a sociálne interakcie, ktoré zahŕňajú prenos rôznych zložitých údajov a multimediálnych objektov. Zabezpečenie obsahu citlivých, ako aj osobných transakcií prostredníctvom otvorených sietí pri súčasnom zabezpečení súkromia informácií sa stalo nevyhnutným, no čoraz náročnejším. Oblasť výskumu informačnej a multimediálnej bezpečnosti preto priťahuje čoraz väčší záujem a rozsah jej aplikácií sa výrazne rozširuje. Mechanizmy zabezpečenia komunikácie boli preskúvané a vyvinuté na ochranu súkromia informácií a multimédií pomocou šifrovania a digitálnej steganografie, ktorá poskytuje dve najzrejmšie riešenia.

Kritériá úspešnosti v steganografii súvisia so zoznamom skôr konkurenčných požiadaviek na: 1) kvalitu stego obrazu; 2) krycia schopnosť; 3) tajná odhaliteľnosť; a 4) odolnosť voči aktívnym útokom. Väčšina existujúcich steganografických schém sa pokúša riešiť iba prvé tri požiadavky, zatiaľ čo robustnosť závisí od aplikácie [13] a väčšina schém zvažuje scenár pasívneho správca, v ktorom správca nezasahuje do súboru stego [12].

Vo všeobecnosti je hlavnou slabinou väčšiny steganografických schém to, že zmeny v štatistike obrázka po vložení údajov môžu byť detekované steganalyzátormi [24]. Schémy skrývania údajov by mali minimalizovať skreslenie vkladania, čím by sa vytvoril vysoko kvalitný stego-obraz, aby odolal útokom steganalýzy [25].

]]

## Kapitola 2

# Skrývanie dát v rámci iných dát

V súčasnosti sa digitálna steganografia delí na niekoľko druhov podľa typu digitálneho objektu, v rámci ktorého sú tajné dáta skrývané. Podkapitola 2.1 priblíži digitálnu steganografiu a pojmy s ňou súvisiace. Zvyšok tejto kapitoly predstavuje jednotlivé druhy steganografie a vytvára prehľad ich metód. Prvým z nich je skrývanie dát v texte (podkapitola 2.2), neskôr v obrázkoch (podkapitola 2.3), vo zvuku (podkapitola 2.4), vo videu (podkapitola 2.5), v sieti (podkapitola 2.6) a nakoniec sú spomenuté ďalšie zaujímavé druhy steganografie (podkapitola 2.7), ktoré naznačujú, kam aktuálny vedecký výskum v tejto oblasti smeruje. V tejto kapitole je vynechaná steganografia spustiteľných súborov, ktorej sa užšie, no detailnejšie venuje nasledujúca kapitola 3.

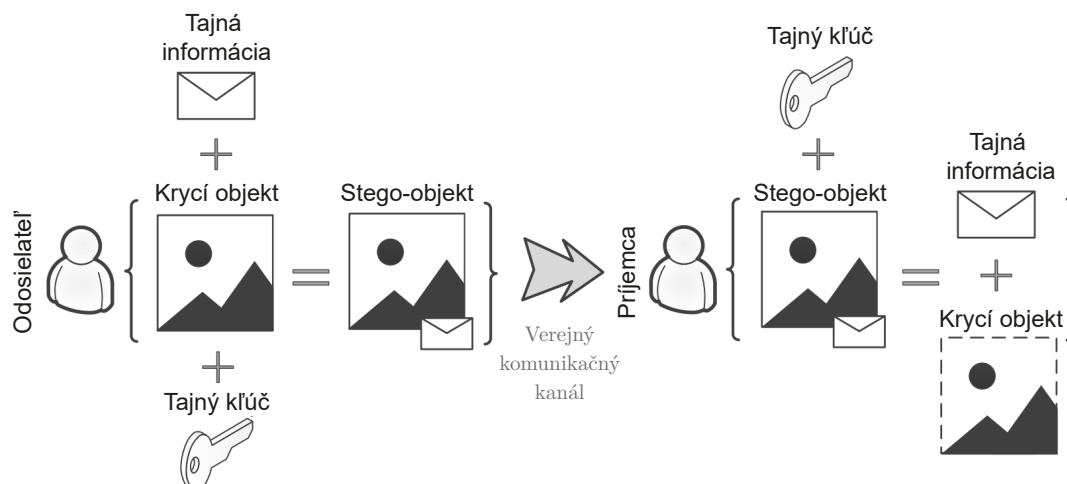
### 2.1 Úvod do digitálnej steganografie

V dnešnom digitálnom svete sa informácie prenášajú internetom častejšie ako kedykoľvek predtým. Preto existencia technológie zabezpečenia a ochrany citlivých a súkromných správ je takmer nevyhnutná. *Digitálna steganografia* (ďalej len *steganografia*) je umenie a veda nenápadnosti ukrytia informácií v skrytých kanáloch, aby sa zabránilo ich odhaleniu. Pojem steganografia pochádza z gréčtiny, kde *steganos* znamená „zakryté“ alebo „skryté“ a *graphein* znamená „písať“. Jej cieľom je skryť informáciu v rámci nosného digitálneho objektu (skrytý kanál) tak, aby nikto okrem odosielateľa a príjemcu netušil o prítomnosti skrytej informácie. [4] [32] [18]

*Steganograf* je osoba, ktorá aplikuje steganografickú metódu. Digitálne objekty v rámci steganografického systému obsahujúce skrytú informáciu sa nazývajú *stego-objekty* (niekedy aj *steganogramy*) a objekty, ktoré ju neobsahujú sa nazývajú *krycie objekty*. Vďaka tomu, že sú tajné informácie skrývané vo vnútri multimediálnych objektov (text, obrázkov, sieťové pakety, ...), môžu byť prenášané otvoreným komunikačným kanálom, keďže vedomosť o existencii skrytej informácie má len odosielateľ a príjemca. Informácie samotné, ako aj krycí objekt, ktorý ich skrýva, môžu byť rozdielnymi digitálnymi objektmi. [4] [36] [12]

Niekedy sa používa aj *steganografický kľúč*, ktorý je tajný. Tento kľúč riadi proces vkladania a extrakcie informácie. Jeho úlohou môže byť napr. rozptýlenie tajnej informácie na podmnožinu všetkých vhodných miest v krycom objekte. Bez kľúča je táto podmnožina neznáma, a teda nie je možné informáciu späťne reprodukovať/extrahovať. Útočník by sa, pri snažení nájsť ukrytú informáciu, dostal len k zmesi použitých a nepoužitých miest v krycom objekte. Obrázok 2.1 znázorňuje steganografický systém, ktorý používa takýto kľúč. [40] [36] [20]





Obr. 2.1: **Proces digitálnej steganografie** – vkladanie a extrakcia tajnej informácie za použitia steganografického kľúča. (Prevzaté a preložené z [12])

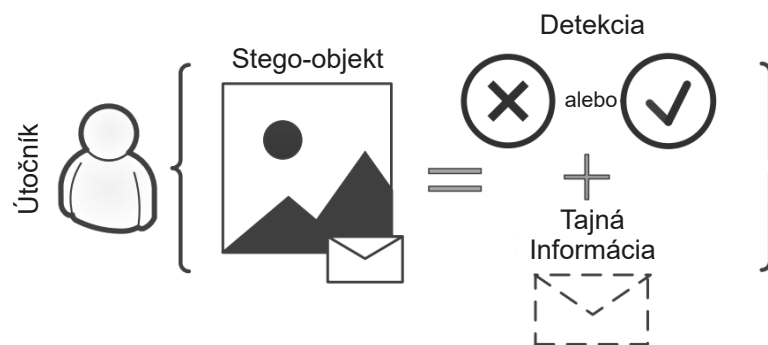
[[Osobitnú pozornosť si zaslúži samotný úvod tejto práce (kapitola 1). Výberom vždy prvého písmena každého odseku vzíde tajná správa v anglickom jazyku: „UNSEEN“. V prípade, že až do tohto momentu čitateľ práce nedostal podozrenie, že tento text by mohol byť stego-objektom, potom sa pokus o použitie jednoduchej steganografie vydaril. Keďže steganografia ako taká počíta s ľudskou naivitou, je relatívne vysoká pravdepodobnosť, že stego-objekt bol detegovaný až v tejto chvíli. Avšak pokiaľ prvý odsek úvodu práce spôsobil akékoľvek pochybenie u čitateľa, snaha o použitie steganografickej metódy sa nepodarila (príklad inej nevhodnej steganografickej metódy znázorňuje obrázok 2.2). Od tejto chvíle je táto práca odhalená ako nosný objekt steganografie, a teda pokus o ukrytie ďalšej tajnej správy v rámci tohto textu by bol riskantný.]]

[4]

random capitalosis is a rare disease often  
contracted by careless internet users. this sad  
illness causes the affected person to randomly  
capitalize letters in a body of text. please  
do not confuse this disease with a blatant  
attempt at steganography.

Obr. 2.2: **Príklad zlej steganografickej metódy** – skrytá informácia je ľahko pozorovateľná napriek tomu, že je asi desaťkrát menšia ako krycí objekt. (Prevzaté z [4])

*Steganalýza* (obrázok 2.3) sa oproti tomu snaží objaviť prítomnosť skrytých informácií. *Steganalytické systémy* sa používajú na zistenie, či digitálny objekt obsahuje tajnú správu. Ide o veľmi náročnú disciplínu, keďže jej úspech je založený na zraniteľnostiach steganografických techník. [11] [32]



Obr. 2.3: Proces steganalýzy (Prevzaté a preložené z [12])

### 2.1.1 Vlastnosti digitálnej steganografie

Pre steganografiu sú definované tri vlastnosti, vďaka ktorým je vhodná na skrývanie informácií: [3] [21] [11]

1. *Nepostrehnuteľnosť* – môže ísť o číslo určujúce kvalitu stego-objektu, ako maximálny pomer signálu k šumu, kde vyššie číslo implikuje vyššiu kvalitu stego-objektu. Ide o naplnenie základnej požiadavky steganografie, a to aby stego-objekt bol ľudským vnemom nerozoznateľný od krycieho objektu tak, aby nevzbudzoval podozrenie. Teda tajná správa musí spôsobiť len nepatrné zmeny krycieho objektu. V prípade odlišných krycích objektov môže byť táto vlastnosť definovaná rôzne.
2. *Robustnosť (odolnosť)* – popisuje odolnosť skrytej informácie voči zmenám krycieho objektu (pridanie náhodného šumu, stratová kompresia, ...). Môže byť vyjadrená číslom, ktoré vznikne podielom kvality neporušeného stego-objektu ku kvalite stego-objektu narušeným steganalytickými zásahmi.
3. *Kapacita* – číslo určujúce maximálny počet bitov tajnej informácie, ktoré je možné ukryť v rámci krycieho objektu.

Literatúra niekedy odčleňuje od vlastnosti nepostrehnuteľnosť ďalšiu vlastnosť – *nedetegovateľnosť*. V tomto prípade ide o odolnosť steganografickej metódy voči detekcii jej použitia za pomoci steganalytických prístupov (napr. štatistické techniky atď.). [13]

### 2.1.2 Bezpečnosť digitálnej steganografie

Vedecké štúdium steganografie odštartoval Gustavus J. Simmons v roku 1983, kedy predstavil klasický steganografický model hovoriaci o plánovaní úteku dvoch väzňov. Tí si vymieňajú správy kontrolované dozorcom, a preto musia tajiť svoje plány v rámci neškodne pôsobiacich (krycích) objektov. Týmto si sú schopní navzájom vymeniť svoje stego-objekty. Tie sú posielané verejným kanálom a dozorca tak môže svojvoľne kontrolovať všetky ich správy. Dozorca môže ku kontrole pristúpiť nasledovne: [32] [1] [16]

- **Aktívne** – dozorca zakaždým pozmení správu od oboch väzňov, aj keď nemusí mať podozrenie, že ide o stego-objekt.
- **Pasívne** – dozorca kontroluje všetky správy a snaží sa zistiť, či ide o stego-objekt, pričom v prípade podozrenia zasiahne.

V digitálnom svete by príkladom aktívneho prístupu dozorcovi mohlo byť pozmenenie správy v podobe rôznych operácií nad vymieňajúcim objektom. Stratová kompresia, konverzia formátu objektu alebo dolnopriepustné filtrovanie sú jednými z možností, ako aktívne kontrolovať digitálne objekty. Avšak väčšina pasívnych dozorcov deteguje stego-objekty analýzou štatistických vlastností správ. [32]

*Bezpečnosť steganografie* (alebo aj *stego-bezpečnosť*) je zaistená, ak je možné zaručiť, že dozorca nie je schopný – aktívnym ani pasívnym útokom – naplniť nasledujúce ciele: [20]

1. *Detegovať krycí objekt ako podozrivý* – cieľ so zameraním na odhalenie existencie tajnej komunikácie z pohľadu krycieho objektu, čo je v súčasnosti východiskovým bodom steganalýzy.
2. *Extrahovať tajnú informáciu* – cieľ so zameraním na odhalenie existencie tajnej komunikácie z pohľadu tajnej informácie. To znamená, že nejde o detekciu stego-objektu ako celku, ale o snahu z neho skrytú informáciu priamo extrahovať. Je to však možné len za podmienok úplného prístupu k potenciálnemu stego-objektu a nejakých počiatočných znalostí o použitej metóde steganografie. Takto je možné priamo určiť prítomnosť steganografie.

Existuje alternatívna definícia, ktorá vyžaduje, aby relatívna entropia<sup>1</sup> medzi stego-objektmi a nezávislými identicky distribuovanými vzorkami z nejakého rozdelenia pravdepodobnosti krycích objektov, bola malá. [16]

### 2.1.3 Útoky steganalytických systémov

Útoky steganalýzy je možné klasifikovať do štyroch úrovní podľa útočníkom nadobudnutých znalostí o steganografickom systéme. Platí, že čím viac znalostí o útočiacom celi môže útočník získať, tým ľahšie dosiahne cieľ útoku – útok vyššej úrovne. Čím vyššej úrovni útoku dokáže steganografický systém odolať, tým vyššiu úroveň stego-bezpečnosti dosahuje. Definujeme nasledujúce úrovne útokov: [20] [11]

1. **Útok na stego-objekt** (angl. *Stego Only Attack – SOA*) – Ide o útok primárnej úrovne, pri ktorom útočník disponuje len potenciálnym stego-objektom. Avšak prístup k pôvodnému kryciemu objektu spreď vloženia tajnej informácie nemá. Útočník môže použiť metódu štatistickej analýzy na modelovanie rozloženia bežne sa vyskytujúcich krycích objektov, čím následne môže detegovať prítomnosť steganografie, alebo iné bežné metódy steganalytických systémov.
2. **Útok známeho krycieho objektu** (angl. *Known Cover Attack – KCA*) – Predstavuje útok druhej úrovne. V tomto útoku má útočník okrem predchádzajúcich znalostí z prvej úrovne (SOA) aj dvojicu pôvodný krycí objekt a jemu zodpovedajúci stego-objekt. Okrem metód z prvej úrovne môže analyzovať práve túto dvojicu a pokúsiť sa prísť na použitý steganografický algoritmus.
3. **Útok vybraného stego-objektu** (angl. *Chosen Stego Attack – CSA*) – Útok tretej úrovne, kde útočník, okrem znalostí z druhej úrovne (KCA), môže praktizovať proces vkladania a extrakcie súčasného steganografického systému a sledovať tak zmeny krycieho objektu, pričom sa snaží spárovať vytvorený stego-objekt s potenciálnym stego-objektom. Ide o znalosť steganografického algoritmu, pričom je však stále steganografický kľúč tajný.

---

<sup>1</sup><https://towardsdatascience.com/information-entropy-c037a90de58f>

4. **Adaptívny útok vybraného stego-objektu** (angl. *Adaptive Chosen Stego Attack* – *ACSA*) – Posledná štvrtá úroveň útokov na steganografický systém, predstavuje opakovaný pokus o úspech v rámci tretej úrovne (CSA).

Úroveň SOA je možné považovať za pasívny útok, pričom ostatné (vyššie) úrovne za aktívny. Všetky tieto úrovne sa sústreďujú na krycie objekty bez akejkoľvek znalosti o tajnej správe, pretože práve krycie objekty sú najčastejšie útočníkom odchytené a analyzované. [20]

#### 2.1.4 Steganografia vs. vodotlač

Utažovanie informácií pomocou steganografie a vkladanie *digitálneho vodoznaku* (ďalej len *vodoznak*) pomocou *digitálnej vodotlače* (ďalej len *vodotlač*) je dobre zavedený a rozvíjajúci sa vedný odbor. Vodotlač je špeciálna forma steganografie, a preto s ňou veľmi úzko súvisí. Medzi jej časté aplikácie patrí: [12] [32]

- označovanie,
- ochrana autorských práv,
- ochrana integrity (neoprávnená manipulácia),
- monitorovanie a
- podmienený prístup.

Pri vodotlači sa skrytá informácia týka nosného objektu a poskytuje o ňom dodatočné informácie alebo sa týka jeho vlastností. Stego-objekt je zároveň primárnym objektom komunikačného kanála a o prítomnosti vodoznaku používateľa vedomosť majú, resp. môžu mať. V steganografii zvyčajne skrytá informácia nijako nesúvisí s nosným objektom, avšak prostredníctvom neho sa skrytá informácia odovzdáva. V tomto prípade je mimoriadne dôležité, aby skrytá správa odhalená nebola, keďže ona samotná je primárnym objektom komunikačného kanála. Pri oboch prístupoch je dôležité zachovať nepostrehnuteľnosť a robustnosť, čo výrazne ovplyvňuje vstavanú kapacitu pre vložené informácie. [32]

Kým steganografia je zraniteľná aj voči pasívnemu útoku, vodotlač môže byť ohrozená len aktívnym. Jej bezpečnosť je prelomená v momente, ak sa útočníkovi podarí vodoznak sfalšovať, zničiť alebo ním manipulovať. [20]

#### 2.1.5 Steganografia vs. kryptografia

Najbezpečnejším spôsobom ako ochrániť informácie v súkromí je transformácia samotných údajov do inej formy. Transformované údaje pochopia len používatelia, ktorí ich dokážu transformovať späť do pôvodnej formy. Takýto spôsob ochrany informácií sa nazýva *šifrovanie* (alebo *kryptografia*). [4] [36]

Hlavnou nevýhodou tohto prístupu je fakt, že existencia údajov nie je tajná. Dáta, ktoré boli zašifrované síce sú nečitateľné, ale stále existujú ako dáta. Široko dostupné šifrovacie algoritmy sú dnes veľmi sofistikované a ich bezpečnosť môže byť preukázaná známymi zložitými matematickými problémami. Preto je veľmi obtiažne takéto dáta dešifrovať používateľom, ktorému neboli adresované. Ak by však zručný používateľ dostal dostatok času a počítačového výkonu, mohlo by sa mu to napokon podať. Riešením tohto problému je práve steganografia. [4] [40] [36]

Zašifrované informácie je ťažšie odlíšiť (v kontexte steganalýzy), na rozdiel od prirodzene sa vyskytujúcich digitálnych objektov (napr. obyčajný text) na nosnom médiu. Preto bezpečnejším variantom v súkromnej/tajnej komunikácii je kombinácia kryptografie a steganografie – viacúrovňové zabezpečenie. Pri použití tajného steganografického kľúča, alebo aj pri jeho kombinácii s kryptografickým kľúčom, by sa mal uplatniť *Kerckhoffsov princíp*. Ten tvrdí, že kľúče predstavujú vstupné parametre pre steganografický algoritmus, a preto je bez ich vedomosti tajná správa v absolútnom bezpečí. To znamená, že samotný steganografický algoritmus je možné odtajniť, lebo len disponovaním dešifrovacieho kľúča je možné rozhodnúť, či načítané bity sú skutočnou ukrytou správou. [40] [36] [20]

### 2.1.6 Začiatky a vývin steganografie

Steganografia a utajovanie informácií nie sú novými praktikami. Prvýkrát bola praktizovaná počas starovekého Grécka, kde sa hovorí o tetovaní oholených hláv poslov (obrázok 2.4). Následne sa počkalo kým im dorástli vlasy a mohli byť poslaní osobne doručiť skrytú správu. Keď posol dorazil k príjemcovi tajnej správy, ten mu oholil hlavu a správu si mohol prečítať. [18] [32]



Obr. 2.4: **Praktika steganografie starovekého Grécka** – tetovanie oholených hláv poslov. (Prevzaté z [24] a [14])

Silu a potenciál steganografických techník ukazujú starovekí Číňania. V čase dynastie Jüan (1280–1368 n. l.) starovekej Číne vládli Mongoli. Slávny príbeh z tohto obdobia hovorí o úspešnom povstaní Číňanov Han. Pri príležitosti každoročného sviatku v polovici jesene, Číňania upiekli tzv. mesačné koláče, pričom sa v týchto krycích predmetoch ukrývala tajná správa o podrobnom pláne útoku. Plánovaná vzburá Číňanov bola vďaka informovanosti skrz koláče napokon úspešná a zvrhla mongolský režim. [32]

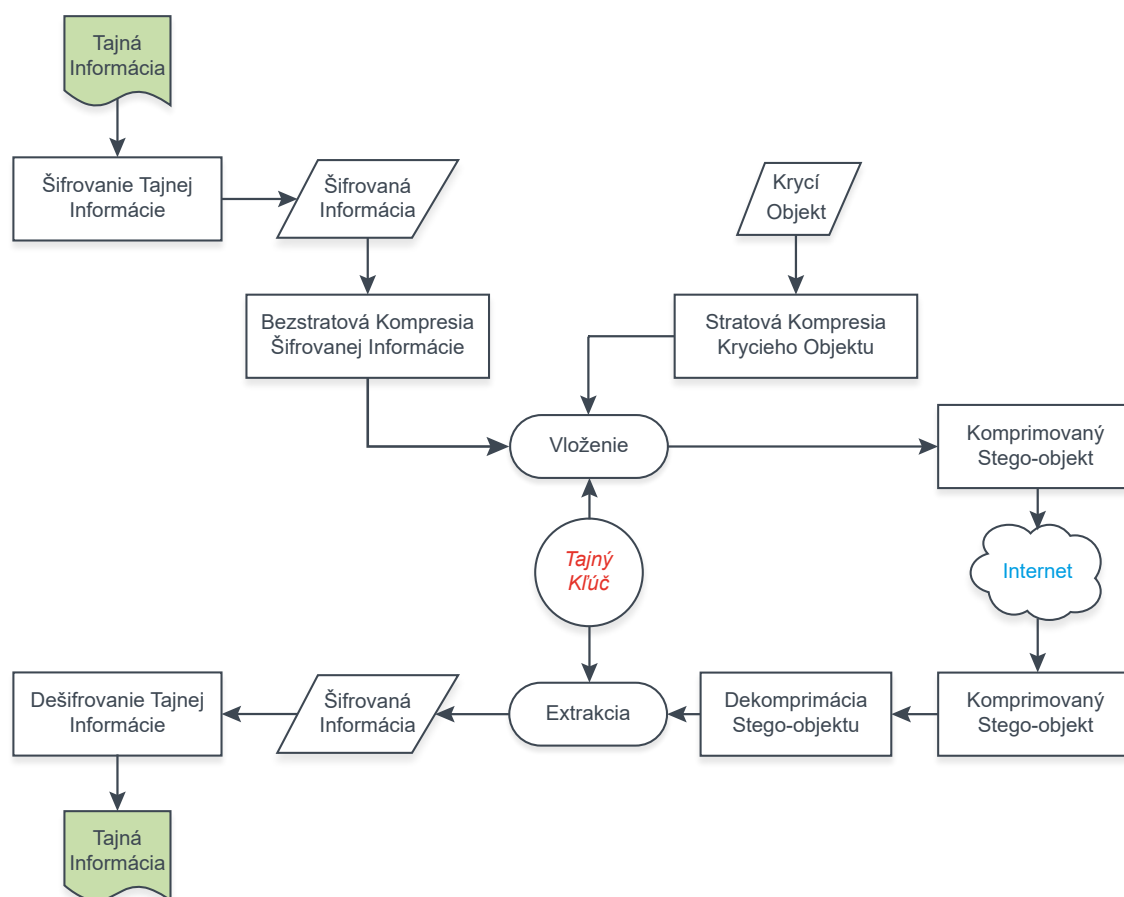
Techniky utajovania informácií sa veľmi spájajú s obdobím druhej svetovej vojny, kedy nacistické Nemecko prišlo v roku 1941 s úplne novou technikou. Mikrobodky spočívali v tom, že tajná správa o veľkosti papiera bola odfotografovaná a zmenšená na veľkosť tlačenej bodky. Týmto spôsobom bolo možné skrývať nielen textové správy ale aj obrázky či kresby. [18] [32]

### 2.1.7 Kompresia dát v steganografii

Úroveň zabezpečenia steganografie je možné zvýšiť použitím techniky, ktorá je dôležitou súčasťou informačnej bezpečnosti. *Kompresia dát* má za úlohu zmenšiť veľkosť digitálneho objektu, čím sa tajná správa ľahko skryje. Dáta sú po skomprimovaní bezpečnejšie a ľahšie sa s nimi manipuluje. Existujú dva typy techník kompresného algoritmu: [39]

- **Bezstratová kompresia** – hľadá dlhé reťazce kódu a vytvorí z nich alternatívne, kratšie reťazce. Nedochádza pri tom k žiadnej strate dát, a preto je možné rekonštruovať komprimované dáta do pôvodnej formy, ktorá je presne rovnaká ako predtým.
- **Stratová kompresia** – hľadá časti kódu, ktoré nie sú primárne zaujímavé pre ľudské vnímanie a odstráni ich. Keďže kompresný pomer tejto techniky je vyšší, bežne sa používa na veľké multimediálne objekty, ktoré je nutné veľkostne zmenšiť (obrázky, videosúbory, ...).

Nasledujúci obrázok 2.5 znázorňuje proces steganografie za použitia kryptografie a kompresie.



Obr. 2.5: **Proces viacúrovňovej steganografie zahrňujúci kryptografiu a kompresiu** – Najprv sa tajná informácia zašifruje a následne skomprimuje. Súbežne s tým sa skomprimuje aj krycí objekt a aplikuje sa steganografia za použitia tajného kľúča. Tento komprimovaný súbor môže byť poslaný cez internet na miesto určenia. Prichádzajúce zakódované bity sú príjemcom dekomprimované, použitím steganografického kľúča sa tajná informácia extrahuje a na záver sa dešifruje. (Prevzaté a upravené z [39])



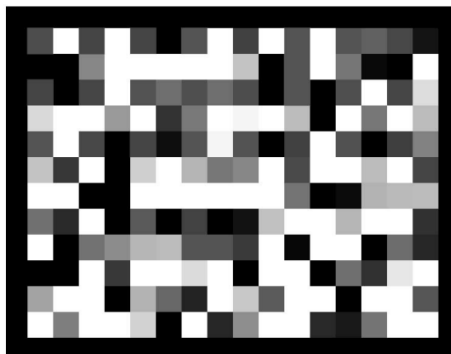
### 2.1.8 Použitie steganografie

Pre použitie steganografie sú najvhodnejšie digitálne objekty s vyšším stupňom *redundancie*. Redundancia je definovaná ako počet bitov, ktoré sú duplicitné alebo navyše od tých, ktoré sú požadované pre nevyhnutne presnú reprezentáciu objektu. Po odstránení redundantných bitov sa objekt nezmení. Digitálne obrázky alebo zvukové nahrávky obsahujú veľa redundantných bitov v podobe šumu, preto sú veľmi dobrými kryciami objektmi. Avšak je žiadúce počítať s tým, že v tomto prípade by mohol byť stego-objekt skomprimovaný, čo by pravdepodobne tajnú správu poškodilo. Výhoda je na strane steganografa, ak dopredu vie, aká technika kompresie sa použije a či vôbec. [32] [40] [1]

### 2.1.9 Limity a obmedzenia digitálnej steganografie

Steganografia a šifrovanie naplňajú osobitné ciele, avšak sú obmedzené rovnakým predpokladom. Predtým ako sa odosielateľ s príjemcom dohodne na komunikácii skrytým kanálom, musia si súkromne dohodnúť metódu steganografie, príp. steganografický kľúč. Rozdiel je v tom, že ak príjemcovi príde šifrovaná správa, okamžite vie, že bola použitá kryptografia, ale pri použití steganografie na to sám pravdepodobne nepríde. Šanca sa znižuje s vyšším počtom krycích objektov, pretože aj keby príjemca vedel o použitej steganografii, nevie kde konkrétne má skrytú informáciu hľadať. [4] [40]

Ďalším obmedzením je *integrita* krycieho objektu, ktorá má výrazný vplyv na jeho kapacitu. Platí, že čím menej obmedzení je pre integritu krycieho objektu, tým vyššiu kapacitu má – potenciál na skrytie údajov. Napr. integrita textu z obrázka 2.2 je obmedzená použitým jazykom a témou samotného textu. Naopak oveľa menšie obmedzenie platí pre integritu obrázka 2.6 pripomínajúceho statický obraz televízie. Podiel informácií, ktorý je možné ukryť do tohto objektu je mnohonásobne vyšší. Jediné, čo by mohlo pri tomto obrázku vzbudzovať pozornosť je jeho výpovedná hodnota. [4]



Obr. 2.6: **Statický obraz televízie** – ukážka nosného objektu steganografie s veľmi nízkym obmedzením integrity. (Prevzaté z [4])

## 2.2 Skrývanie dát v texte

Techniky *textovej steganografie* sú založené na použití písaného prirodzeného jazyka. Takýto digitálny textový súbor ukrýva tajné informácie, čím slúži ako krycí objekt. Ide o asi najťažší typ steganografie, pretože textový súbor obsahuje malú redundanciu dát v porovnaní s obrázkom, videom či zvukom. Navyše, štruktúra textového súboru je totožná s jej

vzhľadom, na rozdiel od iných digitálnych objektov (napr. obrázkov), preto je textová forma dát náchylnejšia na útok. Existujú tri základné kategórie metód: [13] [31] [35]

1. *Metódy založené na formáte* (angl. *Format-Based Methods*)
2. *Náhodné a štatistické metódy* (angl. *Random and Statistics Methods*)
3. *Lingvistické metódy* (angl. *Linguistic Methods*)

V prípade vyššieho záujmu je možné nahliadnuť do prílohy A, kde sú špecifikované vyššie zmienené kategórie metód spolu so základnými príkladmi ich techník. Keďže príloha A približuje len základné techniky textovej steganografie, je tiež možnosťou – pre obsiahnejší prehľad techník – nahliadnuť do literatúry [23].

## 2.3 Skrývanie dát v obrázkoch

Najpopulárnejším druhom steganografie je *obrazová steganografia*, pretože obrázky majú veľkú bitovú redundanciu a ľahko sa šíria internetom. Navyše, takéto metódy často vkladajú informácie ako šum, ktorý je takmer nemožné ľudským zrakom spozorovať.

Vo všeobecnosti možno techniky obrazovej steganografie rozdeliť do dvoch skupín podľa spôsobu ukrytia informácie. Prvá skupina mení obraz a druhá modifikuje formát obrazového súboru. Druhá skupina techník je menej robustná. Hlavnú úlohu v tejto steganografii má kompresia, keďže obrazové súbory sú zväčša veľmi veľké. Preto je potrebné vyvíjať metódy odolné voči takémuto útoku. Metódy modifikujúce obraz sa delia na: [41] [15]

1. *Metódy priestorovej domény* (angl. *Spatial Domain Methods*)
2. *Metódy transformačnej domény* (angl. *Transform Domain Methods*)
3. *Metódy rozprestretého spektra* (angl. *Spread Spectrum Methods*)
4. *Štatistické metódy* (angl. *Statistical Methods*)
5. *Techniky skreslenia* (angl. *Distortion Techniques*)

Druhou skupinou sú techniky zahŕňajúce *vkładanie súborov* a *vkładanie paliet*. Taktiež existujú ďalšie metódy, ktoré pozmeňujú prvky v obraze, a to *techniky generovania obrazu* a *techniky úpravy obrazových prvkov*. Špeciálnym typom techník priestorovej a transformačnej domény je *adaptívna steganografia*. V tejto podkapitole sú priblížené metódy prvej skupiny modifikujúce obrazový súbor. [41] [15]

Pri vyššom záujme sa ponúka možnosť nazrieť do prílohy A. Jej obsahom je špecifikácia aj vyššie spomenutých skupín metód obrazovej steganografie so základným zhrnutím niektorých techník týchto skupín.

## 2.4 Skrývanie dát vo zvuku

Keď je vložená tajná informácia do digitalizovaného zvukového signálu, ide o použitie techniky *zvukovej steganografie* (alebo aj *audiosteganografie*). Takéto vloženie informácie vedie k miernej zmene binárnej sekvencie zodpovedajúceho zvukového súboru (napr. formáty MP3, WAV, ...). Keďže ľudský sluch je oveľa citlivejší ako ľudský zrak, je o to zložitejšie vložiť informáciu do zvukového súboru ako do obrazového. Preto ju tieto metódy vkladajú ako



šum s frekvenciou mimo dosahu ľudského sluchu. Avšak v ich prospech hrá fakt, že zvukové signály majú charakteristickú redundanciu a nepredvídateľný charakter, vďaka čomu sú ideálne pre tajnú komunikáciu. Všeobecne môžeme metódy kategorizovať do dvoch skupín: [5] [28]

1. *Metódy časovej domény* (angl. *Temporal Domain Methods*)
2. *Metódy transformačnej domény* (angl. *Transformation Domain Methods*)

Príloha A je k dispozícii pre prípadný záujem o špecifikáciu skupín metód zvukovej steganografie. Súčasťou prílohy sú aj jej základné techniky.

## 2.5 Skrývanie dát vo videu

Techniky *videosteganografie* využívajú ako krycí objekt videosúbor, ktorý má veľký kapacitný potenciál pre tajné informácie, pretože obsahuje vysoký stupeň priestorovej a časovej redundancie. Navyše, vďaka pokroku v oblasti internetu a multimediálnych technológií, sa videosúbory stali obľúbenými stego-objektmi. Takýto súbor je menej náchylný na steganalýzu, keďže video pozostáva zo série po sebe idúcich a rovnako časovo rozmiestnených statických obrázkov, ktoré môžu byť kombinované so zvukom a textom, do ktorých môžu byť takisto zakódované informácie. [26] [19]

Vo všeobecnosti ide o rozšírenie obrazovej steganografie, a preto je viacero techník obrazovej steganografie použiteľných aj na videá. Keďže je obsah videosúboru dynamický, pravdepodobnosť odhalenia skrytých informácií je nižšia ako pre obrazové objekty. Existuje však veľa efektívnych útokov na videosúbor, ako napr. stratová kompresia, zmena formátu, pridávanie či odstraňovanie snímok počas spracovania videa alebo zmena frekvencie snímok. [30]

Videosteganografiu je možné použiť v rôznych užitočných aplikáciách. Môže ísť o komunikáciu vojenských a spravodajských agentúr, korekciu chýb videa počas prenosu alebo môžu byť metódy tejto steganografie použité aj na prenos dodatočných informácií k videu (napr. titulky) bez potreby zväčšenia šírky pásma. [30]

Techniky videosteganografie sa dajú klasifikovať podľa kompresie na *komprimované* a *nekomprimované* (angl. *raw*) metódy [26]. Iná klasifikácia je založená na doméne vkladania informácie rozlišuje metódy *priestorovej domény* a *transformačnej domény*. Avšak táto podkapitola vychádza z [30], kde je uvedená nasledujúca klasifikácia:

1. *Substitučné metódy* (angl. *Substitution Methods*)
2. *Metódy transformačnej domény* (angl. *Transform Domain Methods*)
3. *Adaptívne metódy* (angl. *Adaptive Methods*)
4. *Metódy založené na formáte* (angl. *Format-Based Methods*)
5. *Metódy generujúce krycí videosúbor* (angl. *Cover Generation Methods*)

Táto klasifikácia je pre prípadných záujemcov zhrnutá a spísaná v prílohe A, kde sa tiež nachádzajú základné príklady techník videosteganografie.

## 2.6 Skrývanie dát v internetovej sieti

Ďalším typom digitálnej steganografie je *sieťová steganografia* (alebo aj *protokolová steganografia*), ktorej sa niekedy hovorí *steganografia 2.0* [19]. Jej techniky využívajú sieťové protokoly a dátové pakety ako krycie objekty. Ich výhodou je ťažká detegovateľnosť paketov obsahujúcich tajnú informáciu. Vo vrstvách modelu ISO/OSI<sup>2</sup> je možné spozorovať viacero skrytých kanálov. Možnosťou je využiť niektoré polia hlavičky TCP/IP paketu alebo iné protokoly transportnej vrstvy (napr. UDP, ICMP, ...). Mohlo by ísť napr. o skrytie informácií do tzv. bloku výplne (angl. *padding*) pre zarovnanie bitov v hlavičke paketu. Keďže žiadne zmysluplné dáta v tejto časti paketu očakávané nie sú, je vysoká pravdepodobnosť, že informácia odhalená nebude. Zaujímavou možnosťou je aj vyvolanie tzv. retransmisie, kedy sa úspešne prijatý paket úmyselne nepotvrdí. Potom tento opakovane prenášaný paket nesie tajné informácie namiesto pôvodných. [30] [4]

## 2.7 Iné možnosti ukrytia dát

Konvenčná steganografia sa zameriava na nepostrehnuteľnosť a nedetegovateľnosť, pretože jej hlavným cieľom je navrhnuť metódy imúnne voči steganálýze. *Nulová-steganografia* (angl. *Zero-steganography*) je vysoko nepostrehnuteľná, nedetegovateľná technika skrývania informácií, pretože počas celého procesu nijako nemodifikuje krycí objekt, a preto je steganálýza absolútne bezpredmetná. Nulová-steganografia je zabezpečená tajným kľúčom, ktorý je vytvorený na základe určitého vzťahu medzi krycím objektom, chaotickou maticou a samotnou tajnou informáciou. Extrakcia tajnej informácie je založená na vzťahu medzi krycím objektom, tajným kľúčom a chaotickou maticou. Okrem nepostrehnuteľnosti a nedetegovateľnosti ponúka táto technika bezpečnosť (zvýšenú najmä vďaka použitiu chaotickej mapy [2]) a dostatočnú kapacitu. [6]

Dôkazom, že vedecký výskum v oblasti steganografie je v plnom prúde je aj nasledujúca myšlienka tzv. *Steganografie s nesprávnym nasmerovaním* (angl. *Misdirection steganography*). Ide o techniku, pri ktorej sú tajné informácie chránené aj v prípade, že útočník má podozrenie o ich vložení do krycieho objektu. Používajú sa dva druhy vkladajúcich tajných informácií do jedného stego-objektu, a to: [25]

- **Skutočné** – sú to informácie, ktoré sú určené na prenos v tajnosti a sú bezpečne skryté v krycom objekte.
- **Falošné/klamlivé** – ide o informácie, ktoré môžu byť útočníkovi známe, pričom tým chránia skutočné informácie.

Avšak vzťah medzi týmito dvomi druhmi informácií žiaden nie je. Cieľom techniky je preniesť skutočné informácie tak, že útočníkovi umožní, aby venoval pozornosť falošným informáciám – tie ho nasmerujú nesprávne. Proces vkladania skutočných tajných informácií závisí od toho, ako vložiť tie falošné. Inštrukcie potrebné k extrakcii falošných tajných informácií (alebo samotné falošné informácie) môžu, ale nemusia, byť potrebné pre extrakciu tých skutočných. Taktiež je možné skryť skutočné tajné informácie v rámci falošných. [25]

---

<sup>2</sup><https://www.techtarget.com/searchnetworking/definition/OSI>

## Kapitola 3

# Digitálna steganografia spustiteľných súborov a ich analýza

Jednou z možností, kam ukryť tajné dáta môže byť aj program. Táto kapitola začína analýzou formátov spustiteľných súborov ELF (podkapitola 3.1) a PE (podkapitola 3.2). Kapitola pokračuje užšou a detailnejšou definíciou spôsobov, ako je možné ukryť tajné dáta v spustiteľných súboroch (podkapitola 3.3). Na záver kapitoly (podkapitola 3.4) sú zhrnuté existujúce software riešenia. **[[Navyše, toto zhrnutie (...), v niektorých prípadoch, zľahka priblíži používané prístupy či techniky.]]**

### 3.1 Analýza formátu ELF

*ELF*<sup>1</sup> definuje štruktúru pre binárne súbory, knižnice a súbory jadra operačného systému (ďalej len OS). Ide o *objektové súbory*, ktoré sú vďaka svojej binárnej reprezentácii spúšťané priamo na procesore. ELF bol pôvodne vyvinutý a publikovaný spoločnosťou UNIX System Laboratories ako súčasť aplikačného binárneho rozhrania (ďalej len ABI<sup>2</sup>). Výbor pre štandardy rozhrania nástrojov TIS<sup>3</sup> zvolil vtedy vyvíjajúci sa štandard ELF ako formát prenosného objektového súboru, ktorý fungoval v prostredí IA-32<sup>4</sup> pre rôzne OS. [38]

Štandard ELF poskytuje vývojárom súbor definícií ABI, ktoré sú prítomné vo viacerých OS. Potrebný počet rôznych implementácií rozhrania sa znižuje, čím sa zníži aj potreba prekódovať a prekompilovať kód. Takýmto zefektívnením vývoja software sa ELF stáva výkonnejší a flexibilnejší binárnym formátom oproti starším a.out a COFF<sup>5</sup>. Existujú tri hlavné typy objektových súborov formátu ELF: [7] [38]

- **Spustiteľný súbor** (angl. *Executable File*) – Spustiteľný súbor obsahuje kód a dáta vhodné na spustenie, tiež špecifikuje rozloženie pamäte procesu.
- **Premiestniteľný súbor** (angl. *Relocatable File*) – obsahuje kód a dáta, ktoré sú vhodné na prepojenie s inými objektovými súbormi na vytvorenie spustiteľného súboru alebo súboru zdieľaného objektu.

---

<sup>1</sup>Spustiteľný a prepojitelný formát (angl. *Executable and Linkable Format – ELF*)

<sup>2</sup>Aplikačné binárne rozhranie (angl. *Application Binary Interface – ABI*)

<sup>3</sup>Štandardy rozhrania nástrojov (angl. *Tool Interface Standards – TIS*)

<sup>4</sup>32-bitová architektúra Intel (angl. *32-bit Intel Architecture – IA-32*)

<sup>5</sup>Formát súboru bežného objektu (angl. *Common Object File Format – COFF*)

- **Súbor zdieľaného objektu** (angl. *Shared Object File*) – známy tiež ako zdieľaná knižnica, obsahuje kód a dáta vhodné na prepojenie v dvoch kontextoch. V prvom ho môže tzv. *linker* spracovať s inými premiestniteľnými súbormi a súbormi zdieľaných objektov na vytvorenie nového objektového súboru. V druhom kontexte ho tzv. *dynamický linker* kombinuje so spustiteľným súborom a inými zdieľanými objektmi, aby vytvoril obraz procesu – reprezentácia spustiteľného súboru v pamäti po tom, čo je do nej načítaná [27].

Formálna špecifikácia zabezpečuje správnosť interpretácie základných strojových inštrukcií OS. Súbor ELF sú zvyčajne výstupom kompilátora alebo linkera, no používajú sa aj pre samotné jadro a moduly jadra OS Linux. [7]

### 3.1.1 Štruktúra súborov ELF

Keďže je formát ELF rozšíriteľný a používa sa pre viacero typov binárnych súborov, ich štruktúra sa mierne líši. Všeobecne pozostáva ELF z dvoch hlavných častí:

1. *Hlavička ELF* (angl. *ELF Header*)
2. *Dáta súboru* (angl. *File Data*)

Dáta súboru je ešte možné rozdeliť na tri menšie časti:

1. *Tabuľka hlavičiek sekcií a sekcie* (angl. *Section Headers Table and Sections*)
2. *Tabuľka hlavičiek programu a segmenty* (angl. *Program Headers Table and Segments*)
3. *Užitočné dáta* (angl. *Payload*)

Existujú dva komplementárne pohľady na súbor ELF (znázorňuje obrázok 3.1). Jeden je použitý pri tvorbe programu linkerom a druhý pri spustení programu, od čoho závisí aj použitie vyššie definovaných hlavičiek. [7] [38]

Z pohľadu linkera	Z pohľadu spustenia programu
Hlavička ELF	Hlavička ELF
Tabuľka hlavičiek programu <i>voliteľné</i>	Tabuľka hlavičiek programu
Sekcia 1	Segment 1
...	Segment 2
Sekcia <i>n</i>	...
...	Tabuľka hlavičiek programu <i>voliteľné</i>
...	
Tabuľka hlavičiek sekcií	

Obr. 3.1: **Komplementárne pohľady na súbor ELF** – Hoci obrázok ukazuje tabuľku hlavičiek programu hneď za hlavičkou ELF a tabuľku hlavičiek sekcií za sekciami, v skutočných súboroch sa to môže líšiť. Okrem toho, sekcie a segmenty nemajú definované poradie. Pevnú pozíciu v súbore má len hlavička ELF. (Prevzaté a preložené z [38])

## Hlavička ELF

V hlavičke *ELF* sa nachádzajú informácie o súbore. Je povinná, pretože zabezpečuje správnu interpretáciu dát počas prepájania (angl. *linking*) a spustenia. Začína sa vždy rovnakými štyrmi bajtmi nazývanými *magická konštanta*. Táto konštanta definuje hexadecimálne formát súboru, pričom sa začína ustáleným prefixom. Magická konštanta vyzerá nasledovne: 7f 45 = **E** 4c = **L** 46 = **F**. Za magickou konštantou sa nachádzajú ďalšie bajty, pričom význam niektorých z nich je vysvetlený vzápätí: [7] [38]

- **Trieda** (angl. *Class*) – bajt nachádzajúci sa hneď za magickou konštantou. Trieda definuje architektúru súboru (0x01 pre 32-bitovú a 0x02 pre 64-bitovú). Súbor využívajúci 64-bitovú architektúru je na výstupe príkazu `readelf` v prostredí Linux označený ako ELF64.
- **Dáta** (angl. *Data*) – Keďže rôzne procesory zaobchádzajú s dátovými štruktúrami a strojovými inštrukciami rôzne, je dôležité informovať procesor o tom, ako interpretovať zostávajúce objekty v súbore. Pre tento účel slúži bajt definujúci dátové pole. Hodnota 0x01 predstavuje LSB<sup>6</sup>, známy aj ako *Little-Endian*. Hodnota bajtu 0x02, naopak, predstavuje MSB<sup>7</sup> známy aj ako *Big-Endian*.
- **Verzia** (angl. *Version*) – číslo verzie ELF formátu. Doposiaľ existuje len jedna verzia, a tak je tento bajt vždy nastavený na hodnotu 0x01.
- **OS/ABI** – Súčasné OS majú veľkú časť ABI spoločnú. Existujú však prípady, kedy sa môžu niektoré OS v ich funkciách líšiť. Tento bajt špecifikuje, aké ABI sa používa, čím dáva najavo, aké funkcie môže OS a aplikácie od súboru ELF očakávať. Ak sa nepoužíva žiadne špeciálne rozšírenie ABI, hodnota bajtu je 0x00 s označením UNIX – System V.
- **Verzia ABI** (angl. *ABI version*) – V prípade potreby, bajt špecifikuje číslo verzie ABI. Ak sa nepoužíva žiadne rozšírenie ABI, hodnota je nastavená na 0x00.
- **Typ** (angl. *Type*) – Dva bajty signalizujúce, o aký typ súboru sa jedná. Môže ísť o spustiteľný súbor (EXEC) – 0x0002, súbor jadra (CORE) – 0x0004 atď.
- **Stroj** (angl. *Machine*) – Dva bajty odhadujúce typ stroja, ktoré určujú architektúru cieľovej inštrukčnej sady (napr. IA-64 – 0x0032, AMD64 – 0x003e, ...).

Skutočná veľkosť niektorých štruktúr objektových súborov je v hlavičke ELF definovaná, preto je možné ich zväčšovať či zmenšovať. Ak sa zmení formát objektového súboru, program môže naraziť na štruktúry, ktoré sú väčšie alebo menšie, ako sa očakávalo. Programy tak môžu ignorovať niektoré informácie. Spracovanie „chýbajúcich“ informácií závisí od kontextu a môže byť špecifikované definovaním rozšírení. [38]

## Tabuľka hlavičiek sekcií a sekcie

*Tabuľka hlavičiek sekcií* (ďalej len *THS*) disponuje informáciami, ktoré popisujú jednotlivé *sekcie* súboru. V THS je zaznamenaná každá z nich, pričom položky THS poskytujú informácie, ako názov sekcie, veľkosť atď. Na špecifických pozíciách sa v THS nachádzajú aj

<sup>6</sup>Najmenej významný bit (angl. *Least Significant Bit* – *LSB*)

<sup>7</sup>Najviac významný bit (angl. *Most Significant Bit* – *MSB*)

záznamy, ktoré sú rezervované a objektový súbor pre nich nemá žiadne sekcie. Významnými dátami o THS – ktoré sú uložené v hlavičke ELF – sú: [38]

- **e\_shoff** – udáva posun (v bajtoch) THS od začiatku objektového súboru.
- **e\_shnum** – udáva počet záznamov (sekcii) THS.
- **e\_shentsize** – udáva veľkosť (v bajtoch) každého záznamu THS.

Sekcie obsahujú väčšinu informácií o objektovom súbore užitočné z pohľadu prepájania objektových súborov (inštrukcie, tabuľku refazcov, tabuľku symbolov, ...). Z tohto dôvodu je táto časť súborov ELF povinná, len ak ide o súbory používané počas prepájania, no pre spustiteľné a iné objektové súbory je len voliteľná. Tiež platí, že každá sekcia má svoju hlavičku, ktorá ju popisuje. Môžu existovať aj hlavičky sekcií, ktoré nemajú žiadnu sekciu. Zároveň, každá sekcia je v pamäti uložená v rámci súvislého bloku bajtov (možno prázdneho). Sekcie sa nesmú prekrývať, teda žiaden bajt sa nenachádza vo viacerých sekciách zároveň. Objektový súbor môže obsahovať aj neaktívne miesta, pretože nie každý bajt súboru musí byť nutne pokrytý THS alebo sekciou. Obsah týchto častí nie je špecifikovaný. [38]

Rôzne sekcie držia riadiace a programové informácie. Súčasťou súboru môžu byť aj tzv. *špeciálne sekcie*, ktoré používa systém a ktoré majú svoje typy a atribúty. Pre ich vysoký počet sa nasledujúci zoznam obmedzuje len na tie najzaujímavejšie: [38] [7]

- **.text** – sekcia obsahuje „text“ v podobe spustiteľných inštrukcií programu. Obsah bude zabalený do segmentu s prístupovými právami na čítanie a vykonávanie. Načíta sa iba raz, pretože obsah sa nezmení.
- **.data** a **.data1** – sekcia obsahuje inicializované dáta s prístupovými právami k čítaniu aj zápisu. Dáta prispievajú k obrazu pamäte programu.
- **.rodata** a **.rodata1** – sekcia obsahuje inicializované dáta, ale s prístupovými právami len k čítaniu. Dáta zvyčajne prispievajú k nezapisovateľnému segmentu v obraze procesu.
- **.bss** – sekcia obsahuje neinicializované dáta (čítanie aj zápis), ktoré prispievajú k obrazu pamäte programu. Podľa definície systém inicializuje dáta s nulami pri spustení programu. Sekcia nezaberá žiaden súborový priestor.
- **.init** – sekcia obsahuje spustiteľný kód, ktorý prispieva k inicializačnému kódu procesu – keď program začne bežať, systém zariadi spustenie tohto kódu pred volaním hlavného vstupného bodu programu (tzv. „main“).
- **.fini** – sekcia obsahuje spustiteľný kód, ktorý prispieva ku kódu ukončenia procesu – keď sa program správne ukončí, systém zariadi spustenie tohto kódu.

Názvy sekcií s prefixom (bodkou) sú vyhradené pre systém, no aplikácie ich môžu používať tiež, ak to pre ne má nejaký význam. Avšak odporúčaním je (pre aplikácie) používať názvy sekcií bez tohto prefixu, aby nedošlo ku konfliktu mien. Objektové súbory formátu ELF umožňujú definíciu vlastných sekcií, pričom názvy sekcií byť unikátne nemusia. Existuje konvencia pre názvy sekcií, ktoré sú určené pre konkrétnu architektúru procesora. V tomto prípade je názov vo vzore *.ARCH.psect*, kde „ARCH“ je používaný názov architektúry v hlavičke ELF (**e\_machine**) a „psect“ je názov sekcie. Potom ide o sekciu „psect“ definovanú architektúrou „ARCH“. [38]

## Tabuľka hlavičiek programu a segmenty

Vo všeobecnosti ide o voliteľnú časť súborov ELF, avšak ak ide o spustiteľné súbory alebo súbory zdieľaného objektu, tie ich obsahovať musia. *Tabuľka hlavičiek programu* (ďalej len *THP*) je primárna dátová štruktúra, ktorá lokalizuje *segmenty* v súbore a obsahuje informácie, ktoré sú potrebné na vytvorenie pamäťového obrazu programu. Jednotlivé segmenty môžu niesť informáciu o zásobníku (segment `GNU_STACK`) či obsluhu výnimiek (segment `GNU_EH_FRAME`). [38] [7]

Každý záznam THP popisuje segment alebo iné informácie, ktoré systém potrebuje na prípravu programu na spustenie. Špecifikácia veľkosti každého záznamu v THP je, podobne ako v časti 3.1.1, uvedená v hlavičke ELF v `e_phentsize` a ich počet (nula alebo viac) je uvedený v `e_phnum`. V hlavičke ELF sa tiež obdobne nachádza miesto začiatku THS (`e_phoff`) v rámci súboru. [38]

Segment objektového súboru pozostáva zo žiadnej alebo z niekoľkých sekcií. O tejto skutočnosti však THP nenesie žiadne informácie. Počet sekcií v rámci segmentu je tiež nepodstatný pre načítanie programu. Avšak prítomné musia byť rôzne informácie na vykonávanie programu, dynamické prepojenie atď. Poradie a zaradenie sekcií v segmente nie je presne definované. Textové segmenty obsahujú inštrukcie a dáta len na čítanie. Na druhú stranu, dátové segmenty obsahujú inštrukcie a dáta s možnosťou zápisu. Spustiteľný súbor rozdelený na segmenty znázorňuje obrázok 3.2. [38]

Pozícia bajtu v rámci súboru	Súbor	Virtuálna adresa
0	Hlavička ELF	
	Tabuľka hlavičiek programu	
	Iné informácie	
0x100	Textový segment	0x8048100
	...	
	0x2be00 bajtov	0x8073eff
0x2bf00	Dátový segment	0x8074f00
	...	
	0x4e00 bajtov	0x8079cff
0x30d00	Iné informácie	
	...	

Obr. 3.2: Príklad spustiteľného súboru ELF architektúry System V – znázornenie štruktúry spustiteľného súboru ELF na architektúre System V vrátane virtuálnych adries a segmentov. (Prevzaté a preložené z [38])

## 3.2 Analýza formátu PE

Formát *PE*<sup>8</sup> definuje súbor špecifikácií, ktorými sa riadia všetky spustiteľné súbory PE, dynamicky pripojené knižnice (DLL<sup>9</sup>) a iné. Ide o štandardný formát týchto súborov rodiny OS Windows. Vychádza zo špecifikácie COFF a bol prijatý spoločnosťou Microsoft

<sup>8</sup>Prenosný spustiteľný súbor (angl. *Portable Executable* – PE)

<sup>9</sup>Dynamicky pripojená knižnica (angl. *Dynamic-Link Library*)



od vydania OS Windows NT 3.1. Odvtedy však prešiel sériou zmien, vďaka ktorým bola pridaná podpora nových funkcií. Základný dizajn formátu PE zostal nezmenený. V súčasnosti existujú dva formáty súborov PE. Pre systémy x86 je to PE32 a pre systémy x64 je to PE32+. [27] [22]

### 3.2.1 Štruktúra súborov PE

Všeobecne sa súbory PE skladajú z *hlavičiek* a *sekcíí*. V súbore sa nachádza hneď niekoľko hlavičiek, ktoré je možné dekomponovať na menšie zmysluplné časti: [8]

1. *Hlavička MS-DOS* (angl. *MS-DOS Header*)
2. *Útržok MS-DOS* (angl. *MS-DOS Stub*)
3. *Hlavička PE* (angl. *PE Header*)
4. *Tabuľka sekcíí* (angl. *Sections Table*)

Hlavička PE sa delí na tri samostatné časti, a to: [8]

1. *Podpis PE* (angl. *PE Signature*)
2. *Hlavička COFF* (angl. *COFF Header*)
3. *Voliteľná hlavička* (angl. *Optional Header*)

Ďalej popisované dátové štruktúry možno pozorovať v hlavičkovom súbore „WINNT.h“, ktorý je súčasťou Windows SDK<sup>10</sup>.

#### Hlavička MS-DOS

*Hlavička MS-DOS* (dátová štruktúra *Image\_MS-DOS\_Header*) je povinnou časťou každého spustiteľného súboru PE. Najdôležitejšie sú dve položky:

- **e\_magic** – Magická konštanta zaberajúca prvé dva bajty hlavičky, ktorá je vždy nastavená na hodnotu `0x4d5a` = **MZ**. *MZ* je jedinečný identifikátor a predstavuje Marka Zbikowského, tvorca MS-DOS. V spustiteľných súboroch sa nachádza od OS MS-DOS a dodnes sa používa kvôli spätnej kompatibilite s MS-DOS.
- **e\_lfnew** – Posledná položka hlavičky MS-DOS (od bajtu `0x3c` v rámci súboru) nesie adresu začiatku hlavičky PE. V skutočnosti patrí táto adresa podpisu súboru PE. Je to nutné z dôvodu, že medzi hlavičkou MS-DOS a PE sa ešte nachádza tzv. *útržok MS-DOS*. Adresa je však zadaná nepriamo, a to ako poradie bajtu od začiatku súboru PE.

Všetky ostatné polia nie sú pri analýze súborov PE také užitočné, pretože len pomáhajú pri spúšťaní programov. [22] [8]

---

<sup>10</sup>Balíček na vývoj software (angl. *Software Development Kit* – *SDK*)



## Útržok MS-DOS

Ako už bolo vyššie načrtnuté, nasleduje útržok MS-DOS. Ide o skutočný program, ktorý spúšťa systém MS-DOS pri načítaní spustiteľného súboru. Pre neskoršie OS Windows sa tu nachádza útržok programu MS-DOS, ktorý beží namiesto skutočnej aplikácie. Zvyčajne táto sekcia len vypisuje buď „Tento program sa nedá spustiť v režime MS-DOS.“, alebo „Tento program musí byť spustený pod win32.“. Táto sekcia je plne zodpovedná za správanie programu, ak by bol spustený v systéme MS-DOS a vytvorený pre spätnú kompatibilitu. Útržok MS-DOS sa nachádza bezprostredne za 64 bajtovou hlavičkou MS-DOS. Je možné tiež vytvoriť vlastný obsah tejto časti súboru. [8] [22]

## Podpis PE

Spustiteľným súborom systému Windows a OS/2 bol pridaný tzv. *podpis PE*, ktorý určuje zmýšľaný cieľový OS (OS/2 alebo MS-DOS, alebo Windows NT). Ak ide o formát súboru PE v systéme Windows NT, podpis PE sa nachádza bezprostredne pred štruktúrou hlavičky PE. Naopak, vo verziách Windows a OS/2 je podpis prvou položkou hlavičky PE. V systéme Windows NT zaberá podpis štyri bajty a má tvar: 0x50 = **P** 0x45 = **E** 0x00 = \0 0x00 = \0. [22]

## Hlavička COFF

*Hlavička COFF* (dátová štruktúra `IMAGE_FILE_HEADER`) disponuje len informáciami vyššej úrovne, ktoré hovoria, ako so súborom PE zaobchádzať. Medzi informáciami sú aj dáta, či ide o dynamickú analýzu alebo spustiteľný súbor a či jeho obraz v pamäti podporuje náhodnú základnú adresu (angl. *Randomized Base Address*). `IMAGE_FILE_HEADER` sa skladá z týchto položiek: [22] [27]

- **Machine** – Dvojбайtový identifikátor reprezentujúci architektúru procesora (napr. 0x8664 pre AMD64, 0x14c pre IA-32, ...).
- **NumberOfSections** – Definuje počet hlavičiek sekcií a tiel sekcií v súbore PE, pričom existuje obmedzenie na 96 sekcií. Pomocou tohto počtu je možné zistiť celkovú veľkosť tabuľky sekcií, keďže každá hlavička sekcie a telo sekcie sú v súbore usporiadané postupne.
- **TimeStamp** – predstavuje dátum vytvorenia súboru PE.
- **PointerToSymbolTable** – obsahuje posun v bajtoch k tabuľke symbolov. Používa sa zriedka (zvyčajne vyplnené nulami), keďže táto informácia je zastaralá.
- **SizeOfOptionalHeader** – definuje veľkosť voliteľnej hlavičky.
- **Characteristics** – príznak (angl. Flag), ktorý predstavuje niektoré charakteristiky súboru pomocou preddefinovaných konštánt (napr. či ide o spustiteľný súbor, systémový súbor, informácia o výskyte ladiacich informácií v súbore, ...).

## Voliteľná hlavička

Napriek názvu, *voliteľná hlavička* (dátová štruktúra `IMAGE_OPTIONAL_HEADER`) je povinnou súčasťou súborov PE pre ich spustenie. Začína hneď za hlavičkou COFF a poskytuje

podrobnejšie informácie o spustiteľnom obraze súboru PE. Má dva hlavné typy podľa architektúry systému (PE32 alebo PE32+). Tento typ je uložený magickou konštantou v prvých 22 bajtoch (0x10b pre PE32 a 0x20b pre PE32+). Výhodou dizajnu tejto hlavičky je, že jej veľkosť nie je pevná. Je určená v hlavičke COFF, čo uľahčuje jej rozšírenie v budúcich úpravách formátu PE. Položky hlavičky, ktoré stoja za zmienku: [8] [27]

- **MajorLinkerVersion** – Predstavuje číslo hlavnej verzie linkera.
- **MinorLinkerVersion** – Predstavuje číslo vedľajšej verzie linkera.
- **SizeOfCode** – Veľkosť kódovej časti v bajtoch. Ak sa kód rozprestiera medzi viacero sekcií potom položka definuje súčet všetkých sekcií kódu.
- **SizeOfInitializedData** – Veľkosť inicializovanej dátovej sekcie v bajtoch alebo súčet všetkých takýchto sekcií, ak existuje viacero inicializovaných dátových sekcií.
- **SizeOfUninitializedData** – Veľkosť neinicializovanej dátovej sekcie v bajtoch. V prípade viacerých neinicializovaných dátových sekcií, súčet všetkých týchto sekcií.
- **ImageBase** – Preferovaná základná adresa v adresnom priestore procesu, na ktorý sa má priradiť spustiteľný obraz. Predvolená hodnota linkera je 0x00400000, no je možné ju zmeniť.
- **SizeOfImage** – Definuje veľkosť adresného priestoru, ktorý sa má rezervovať pre načítaný spustiteľný obraz.

Veľmi dôležitá je tiež posledná položka voliteľnej hlavičky, a to *tabuľka dátových adresárov*. Počet dátových adresárov nie je stanovený a určuje ho samotná tabuľka. Príkladmi dátových adresárov sú: [27]

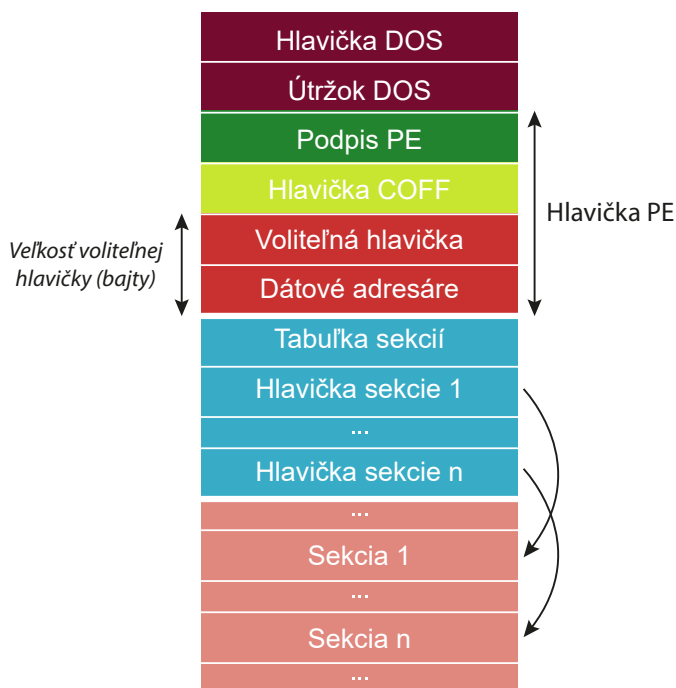
- *Importná tabuľka* (angl. *Import Table*) – Deklaruje dynamické knižnice (DLL) a funkcie, ktoré je potrebné načítať spolu so spustiteľným súborom.
- *Exportná tabuľka* (angl. *Export Table*) – Obsahuje funkcie, ktoré spustiteľný súbor sprístupňuje iným programom na použitie.
- *Tabuľku certifikátov* (angl. *Certificate Table*) – Obsahuje digitálne podpisy vývojára spustiteľného súboru.

### Tabuľka sekcií a sekcie

Nasledujú *hlavičky sekcií*, ktoré sú uložené v tabuľke. Táto tabuľka sa musí nachádzať bezprostredne za voliteľnou hlavičkou (ak existuje), pretože informácia o jej umiestnení nie je nikde uložená. Jej pozícia je daná výpočtom prvého bajtu po hlavičkách. Každý záznam v tabuľke – o veľkosti 40 bajtov – definuje *sekciiu*, ktorá tvorí súvislú pamäťovú oblasť obrazu. Tá je buď neinicializovaná, alebo vyplnená časťami spustiteľného súboru. Sekcie obsahujú obsah súboru vrátane kódu, údajov, zdrojov a iných spustiteľných informácií. Každá sekcia má hlavičku a telo. Hlavičky sekcií majú presne definovanú štruktúru, no telám sekcií pevná štruktúra chýba. Sekcie môžu byť usporiadané takmer akokoľvek, pokiaľ hlavička obsahuje dostatok informácií o príslušnom tele sekcie. Súbor PE pre OS Windows NT má zvyčajne tieto preddefinované sekcie: [22] [27]

- **.text** – Obsahuje kód programu.
- **.bss** – Obsahuje neinicializované dáta vrátane všetkých premenných deklarovaných ako statické v rámci funkcie alebo zdrojového modulu.
- **.rdata** – Obsahuje dáta len na čítanie (reťazce, konštanty a informácie o adresári ladenia).
- **.data** – Obsahuje globálne premenné aplikácie či modulu.
- **.rsrc** – Obsahuje zdrojové informácie pre modul. Dáta sekcie sú štruktúrované do stromu zdrojov.
- **.edata** – Obsahuje informácie o funkciách exportovaných z knižnice DLL. Ak je sekcia k dispozícii, obsahuje exportný adresár na získanie informácií o exporte.
- **.idata** – Obsahuje informácie o funkciách importovaných spustiteľným súborom alebo knižnicou DLL vrátane importného adresára a tabuľky názvov adres importu.
- **.debug** – Obsahuje informácie o ladení, ktoré sú spočiatku umiestnené v tejto sekcii. Ako prostriedok na zhromažďovanie informácií o ladení na jednom mieste môžu byť pri formáte PE aj samostatné súbory (s príponou .DBG). Aj keď táto sekcia obsahuje informácie o ladení, adresáre ladenia sa nachádzajú v časti .rdata uvedenej vyššie. Každý z týchto adresárov odkazuje na informácie o ladení v tejto sekcii.

Obrázok 3.3 znázorňuje štruktúru spustiteľného súboru PE tvoriaceho vyššie rozobrané komponenty.



Obr. 3.3: Štruktúra spustiteľného súboru PE

### 3.3 Spôsoby ukrytia dát v spustiteľných súboroch

Nasledujúce metódy boli vyvinuté pre steganografické vloženie informácií do binárnych súborov programov x86.

#### 3.3.1 Substitúcia inštrukcií

môžeme vkladať informácie pomocou funkčne ekvivalentných inštrukcií (t. j. inštrukcií strojového kódu i386). Na určenie dostupnej kapacity analyzujeme binárne súbory niekoľkých distribúcií operačného systému (OpenBSD 3.4, FreeBSD 4.4, NetBSD 1.6.1, Red Hat Linux 9 a Windows XP Professional). Naše testy ukazujú, že dostupná kapacita vzhľadom na sady ekvivalentných inštrukcií, ktoré v súčasnosti používame, je približne 1/110 bitov (t. j. môžeme zakódovať 1 bit informácie na každých 110 bitov programového kódu). — Všimnite si, že rozlišujeme medzi celkovou veľkosťou programu a veľkosťou kódu. Celková veľkosť programu zahŕňa okrem sekcií kódu aj rôzne sekcie údajov, premiestnenia a BSS. Experimentálne sme zistili, že časti kódu zaberajú v priemere 75 % celkovej veľkosti spustiteľných súborov. Napríklad staticky prepojený spustiteľný súbor s veľkosťou 210 KB obsahuje približne 158 KB kódu, do ktorého môžeme vložiť 1,44 KB (11 766 bitov) údajov.

diskutujeme o vylepšeniach, ktoré môžu viesť k rýchlosti kódovania 1/36.

Na zakódovanie správy používame inherentnú redundanciu v inštrukčnej sade stroja (napr. inštrukčnej sade procesora i386), pretože niekoľko inštrukcií môže byť vyjadrených viac ako jedným spôsobom. Napríklad pridanie hodnoty 50 do registra `eax` môže byť vyjadrené ako „pridať `%eax`, 50“ alebo „`sub %eax, -50`“. Pomocou týchto dvoch alternatívnych foriem môžeme zakódovať jeden bit informácie vždy, keď dôjde k sčítaniu alebo odčítaniu v spustiteľnom kóde. Ďalším príkladom je XORing registra proti sebe, aby sa vymazal jeho obsah: odčítanie registra od seba má rovnaký účinok.

steg86 využíva jednu zo zvláštností kódovania x86: pole R/M z bajtu ModR/M: 7 6 5 4 3 2 1 0 ————— | MOD | REG | R/M | ————— Bajt ModR/M sa bežne používa na podporu variantov registra do pamäte aj z pamäte do registra toho istého pokynu. Napríklad inštrukcia ‘MOV’ má nasledujúce varianty (okrem mnohých iných): | operačný kód | mnemotechnická pomôcka | ————|————— | "89 /r ‘MOV r/m32,r32’ | | "8B /r ‘MOV r32,r/m32’ | Pretože pole ModR/M môže kódovať buď operáciu adresovania pamäte alebo holú register, operačné kódy, ktoré podporujú operácie register-to-memory a memory-to-register tiež podpora viacerých kódovaní operácií medzi registrami. Napríklad ‘mov `eax, ebx`’ možno zakódovať ako buď ‘89 d8’ alebo ‘8b c3’ bez akejkoľvek sémantiky zmeny. To nám dáva jeden bit informácií na duplikovanú sémantiku inštrukcie. Dané dosť inštrukcie register-to-register s viacerými kódovaniami, môžeme s nimi skryť celé správy bitov. Navyše, pretože tieto sémanticky identické kódovania majú často rovnakú veľkosť, môžeme upraviť existujúce binárne súbory bez toho, aby sme museli opravovať premiestnenia alebo adresovanie súvisiace s RIP.

Niekedy môžeme zakódovať viac ako jeden bit na inštrukciu použitím čo najväčšieho počtu ekvivalentných inštrukcií, pretože pri použití sady  $n$  funkčne ekvivalentných inštrukcií môžeme vložiť  $\log_2(n)$  bitov. Pre množinu štyroch inštrukcií môže byť akákoľvek inštrukcia v tejto sade použitá na vloženie dvoch bitov dát. Sady, ktoré sme našli, zvyčajne obsahujú dve alebo štyri inštrukcie. V dvoch prípadoch sa nám podarilo nájsť sedem inštrukcií.

### 3.3.2 Zmena poradia inštrukcií

### 3.3.3 Metóda založená na rozložení kódu

**[[ako sa dajú jednotlivé metódy detegovať, resp. zraniteľnosť metód na detekciu MOŽE TO BYŤ PRI METÓDACH - SPOMENUT.. tiež spomenut distribúciu inštrukcií ekvivalenčných tried - uvedené pred further discussion v Hydan article a s tým spojená detegovateľnosť]]**

## 3.4 Existujúce aplikačné nástroje

V súčasnosti je voľne dostupné len obmedzené množstvo aplikácií pre steganografiu spustiteľných súborov, na rozdiel od iných druhov steganografií. Menej zdokumentované staršie aplikačné nástroje, ktoré sú použiteľné za účelom ukrytia dát v rámci spustiteľných súborov sú:

- **Clotho**<sup>11</sup> – Lahko použiteľný starší nástroj pre OS Windows, ktorý dokáže skryť citlivé dáta do obrázkov, zvuku, videa, spustiteľných súborov a ďalších rôznych formátov súborov (napr. ZIP, RAR, EXE, DLL, ...). Aplikácia umožňuje šifrovanie skrývaných dát alebo komprimáciu výsledného súboru za účelom zníženia veľkosti. Taktiež je možné skryté dáta extrahovať. Navyše, aplikácia ponúka reštrukturalizáciu dát, aby sa skryté dáta dali extrahovať aj použitím archivačného nástroja WinRAR.
- **StegoStick**<sup>12</sup> – Ide taktiež o starší nástroj, ktorý ponúka možnosť skryť akýkoľvek formát dát do obrázku, zvuku, videa alebo súboru EXE, PDF atď. Podporuje tiež šifrovanie.

Nasledujúce tri aplikačné nástroje sa sústreďujú na inštrukčnú sadu x86 kompatibilnú s architektúrou procesorov IA-32, podobne ako metódy uvedené v predchádzajúcej podkapitole 3.3.

### Hydan

*Hydan*<sup>13</sup> je prvým skutočným pokusom – podľa dostupných informácií – o návrh a implementáciu metódy určenej pre steganografiu spustiteľných súborov. Steganografický nástroj implementuje metódu substitúcie funkčne ekvivalentných inštrukcií strojového kódu (vysvetlená v sekcii 3.3.1). Software bol zhodnotený vývojármi jedinou metrikou, a to schopnosťou vkladania bitov za určitý počet bitov krycieho programu. Táto metrika bola ohodnotená hodnotou  $frac{1110}$ , čo implikuje vloženie jedného bitu informácie na každých približne 110 bitov krycieho programu.

### ARMaHYDAN

*ARMaHYDAN*<sup>14</sup> je aplikačný nástroj, ktorý manipuluje s tzv. *voliteľnými bitmi* v inštrukciách procesora ARM. Názov tohto software bol odvodený od názvu vyššie spomenutého

<sup>11</sup><https://www.softpedia.com/get/Security/Encrypting/Nugraha-Clotho.shtml>

<sup>12</sup><https://sourceforge.net/projects/stegostick/>

<sup>13</sup>Oficiálna web stránka (<http://www.crazyboy.com/hydan/>) tohto software a jeho dokumentácie nie je v čase písania tejto práce dostupná.

<sup>14</sup><https://github.com/XlogicX/ARMaHYDAN>

nástroja Hydan. Tento steganografický nástroj implementuje tzv. metódu *Metódu k šílenstvu* (angl. *Method to the Madness*).

[[Tento nástroj sa však zameriava na bity v zátvorkách; 0 bitov v bitoch 19-16. Zdá sa, že tieto bity sú „voliteľné“ (nedokumentované). Zdá sa, že zmena týchto bitov vôbec nezmení činnosť inštrukcie. Disassembler však zvyčajne dekoduje tieto inštrukcie ako NEDEFINOVANÉ (aj keď sa stále vykonávajú bez problémov).

Niektoré inštrukcie majú viac ako 4 voliteľné bity. Niektoré inštrukcie majú tieto bity rozptýlené po celej inštrukcii. Tieto bity nie sú vždy 0, len sú náhodou v príklade MOV.

Mal by som tiež poznamenať, že hoci to nie je zdokumentované, ak sa pozorne pozriete na to, ako sú pokyny zakódované, existuje určitá metóda na šílenstvo. Porovnanie inštrukcie ODCÍTAŤ a POROVNAŤ to veľmi dobre ilustruje. Všimnite si, že inštrukcia CMP robí všetko, čo robí odčítanie, len neukladá výsledok odčítania. Na základe výsledkov SUB alebo CMP by sa nastavili niektoré príznaky (aby k tomu skutočne došlo pre SUB, bit 'S' by musel byť nastavený na '1', pri CMP je to štandardne '1'). Takže porovnanie nižšie uvedených inštrukcií: Bity 21-24 sú relevantné bity označujúce použitú inštrukciu, bit 24 je jediný rozdiel. V prípade týchto inštrukcií odpočítavame Rm od Rn, výsledok by sa dostal do Rd. Tu robím krok späť, aby som ocenil eleganciu ARM. Štruktúra-/kódovanie týchto súvisiacich pokynov je neuveriteľne podobná. V skutočnosti také podobné, že všetko okrem bitu 24 (rozdiel medzi SUB/CMP) a voliteľne bitu 20 je identické. V inštrukcii CMP je len akýsi duch pre Rd. Všimnite si, že tieto bity nemusia byť 0, môžu to byť čokoľvek; keďže sa do Rd aj tak nič nezapisuje.

Hoci vyzerajú rovnako, všimnite si, že druhá verzia je hlúpa. Je dosť hlúpe dať vám:

Program terminated with signal SIGILL, Illegal instruction.

ARMaHYDAN sa snaží šikovne využiť skutočnosť, že tieto bity sú tak tvarovateľné.]]

Avšak, ako poznamenali samotný tvorcovia tohto software, tento spôsob ukrytia dát je príliš zraniteľný a neodporúča pri skrývaní dôležitých dát. Je tomu tak preto, že voliteľné bity inštrukcií sú za predvolených podmienok konzistentné, a preto akákoľvek ich odchýlka je príliš nápadná.

## steg86

Novším a modernejším software je **steg86**<sup>15</sup>. Táto steganografická aplikácia je určená pre binárne súbory architektúry x86 a AMD64. Software je schopný skryť tajné dáta do spustiteľného (binárneho) súboru bez ohľadu na formát (ELF, PE, Mach-O, raw, ...). Výhodou nástroja je, že nemá žiaden vplyv na výkon ani veľkosť krycieho súboru. Steganografický nástroj implementuje metódu, ktorá bola predstavená vývojármi nástroja Hydan, pričom sa tvorcovia steg86 odvolávajú na to, že ich objav bol napísaný úplne nezávisle.

<sup>15</sup><https://github.com/woodruffw/steg86>

## Stilo

Asi najmodernejším riešením pre steganografiu spustiteľných súborov je software *Stilo*<sup>16</sup>, ktorý posunul celý vedecký výskum tejto steganografie o veľký kus dopredu. Boli v ňom predstavené dve nové techniky popísané v sekcii 3.3.2 a 3.3.3. Taktiež bola implementovaná zlepšená metóda nástroja Hydan. Na záver boli identifikované možné zraniteľnosti týchto metód.

---

<sup>16</sup><https://www2.cs.arizona.edu/~collberg/Teaching/620/2008/Assignments/tools/stilo/index.html>

## Kapitola 4

### Návrh ...

#### 4.1 Architektúra aplikácie

Proces vloženia informácie

Proces extrakcie informácie

#### 4.2 Popis zvolenej steganografickej metódy

#### 4.3 Popis skúmania vlastnej modifikácie substitučných tried

moje použitie NOPov – rozsireníe

možné použitie nedokumentovaných NOPov

nahrada NOPov za ine instrukcie ktore nic nerobia

size changing tricks

prefixy retazcovych instrukcii

#### 4.4 Mechanizmus vyhodnocovania vlastností použitých steganografických metód



Kapitola 5

Implementácia ...

## Kapitola 6

### Testovanie a experimenty ...

## Kapitola 7

### Záver

[[PEKNE ZHODNOTENIE STEGANO AKO PRAKTIKY - ODPOVED NA TO CI MA VYZNAM ABY SA VYVIJAL AJ STEGANO AJ STEGANALYZA

V poslednej dobe je veľký záujem o digitálnu steganografiu, teda o ukrývanie tajných správ v komunikácii medzi počítačmi. Tento záujem je zjavne podporovaný zvýšeným množstvom komunikácie, ktorú sprostredkujú počítače a pripojenia k potenciálnym komerčným aplikáciám: Skryté informácie by mohli byť potenciálne použité na odhalenie alebo obmedzenie neoprávneného šírenia nevinne vyzerajúcich „nosičov“ údajov. Z tohto dôvodu sa objavilo množstvo návrhov protokolov na skrytie údajov v kanáloch obsahujúcich obrázky [4], [5], video [5], [6], [7], zvuk [8], [9] a dokonca vysádzať text [10]. Mnohé z týchto protokolov sú mimoriadne chytré a vo veľkej miere sa spoliehajú na vlastnosti týchto kanálov špecifické pre doménu. Na druhej strane, literatúra o steganografii obsahuje aj veľa šikovných útokov, ktoré odhaľujú používanie takýchto protokolov. V dôsledku toho nie je z tejto práce jasné, či je bezpečná steganografia vôbec možná. [16]]

[[NAVRHY DO BUDUCEHO VYSKUMU V STEGO-BEZPECNOSTI

Okrem toho existujú ďalšie dva pohľady na budúci výskum stego-bezpečnosti: 1. Zlepšiť distribúciu kľúčov ďalším zavedením mechanizmu systému verejného kľúča do steganografie. 2. Dosiahnuť kontrolu integrity tajnej správy, ako aj autentizáciu zdroja skrytej správy. [20]]

[[Steganografia môže tiež zvýšiť súkromie jednotlivca. Hoci nejde o náhradu za šifrovanie, digitálna steganografia poskytuje prostriedky súkromnej komunikácie. Samozrejme, je to účinné iba ak sa skrytá komunikácia nezistí. Ak chce človek jednoducho komunikovať bez toho, aby bol vystavený monitorovacím systémom svojho zamestnávateľa, potom je digitálna steganografia dobrá riešenie – najsúkromnejšia komunikácia je tá, ktorá nikdy neexistovala! [4]

MOJA VETA: Steganografia zvyšuje súkromie jednotlivca, pretože najsúkromnejšia komunikácia je taká, ktorá nikdy neexistovala.]]

[[DO ZAVERU - PEKNE SLOVA

Steganografia, ako už bolo spomenuté, skôr vylepšuje ako nahrádza šifrovanie. Správy nie sú bezpečné len preto, že sú skryté. Rovnako tak steganografia nie je o tom, aby sa vaša správa neprezradila – ide o to, aby sa o jej existencii nevedelo. S ohľadom na tieto body položím poslednú otázku: Je v tomto článku ešte nejaké skryté posolstvo? [4]]



# Literatúra

- [1] ANDERSON, R. a PETITCOLAS, F. On the limits of steganography. *IEEE Journal on Selected Areas in Communications* [online]. IEEE. Máj 1998, zv. 16, č. 4, s. 474–481, [cit. 2022-01-24]. DOI: 10.1109/49.668971. ISSN 1558-0008. Dostupné z: <https://ieeexplore-ieee-org.ezproxy.lib.vutbr.cz/document/668971>.
- [2] ANEES, A., SIDDIQUI, A. M., AHMED, J. et al. A technique for digital steganography using chaotic maps. *Nonlinear Dynamics* [online]. Springer. Marec 2014, zv. 75, č. 4, s. 807–816, [cit. 2022-01-27]. DOI: 10.1007/s11071-013-1105-3. ISSN 1573-269X. Dostupné z: <https://link.springer.com/article/10.1007/s11071-013-1105-3>.
- [3] ANTONIO, H., PRASAD, P. W. C. a ALSADOON, A. Implementation of cryptography in steganography for enhanced security. *Multimedia Tools and Applications* [online]. Springer. Máj 2019, zv. 78, č. 23, s. 32721—32734, [cit. 2022-01-26]. DOI: 10.1007/s11042-019-7559-7. ISSN 1573-7721. Dostupné z: <https://link.springer.com/article/10.1007/s11042-019-7559-7>.
- [4] ARTZ, D. Digital Steganography: Hiding Data Within Data. *IEEE Internet Computing* [online]. IEEE. Jún 2001, zv. 5, č. 3, s. 75–80, [cit. 2022-01-06]. DOI: 10.1109/4236.935180. ISSN 1941-0131. Dostupné z: <https://ieeexplore.ieee.org/document/935180>.
- [5] BHATTACHARYYA, S., BANERJEE, I. a SANYAL, G. A Survey of Steganography and Steganalysis Technique in Image, Text, Audio and Video as Cover Carrier. *Journal of Global Research in Computer Science* [online]. Citeseer. Apríl 2011, zv. 2, č. 4, s. 1–16, [cit. 2022-01-28]. ISSN 2229-371X. Dostupné z: <https://www.rroij.com/open-access/a-survey-of-steganography-and-steganalysis-technique-in-image-text-audio-and-video-as-cover-carrier-1-16.php?aid=37026>.
- [6] BILAL, M., IMTIAZ, S., ABDUL, W. et al. Chaos based Zero-steganography algorithm. *Multimedia Tools and Applications* [online]. Springer. September 2014, zv. 72, č. 2, s. 1073–1092, [cit. 2022-01-27]. DOI: 10.1007/s11042-013-1415-y. ISSN 1573-7721. Dostupné z: <https://link.springer.com/article/10.1007/s11042-013-1415-y>.
- [7] BOELEN, M. The 101 of ELF files on Linux: Understanding and Analysis. *Linux Audit: The Linux security blog about Auditing, Hardening, and Compliance* [online]. Máj 2019 [cit. 2022-03-05]. Dostupné z: <https://linux-audit.com/elf-binaries-on-linux-understanding-and-analysis/>. Path: Home; The 101 of ELF files on Linux: Understanding and Analysis.
- [8] DEMIDENKO, M., KAIPIYEV, A., NASERI, M. V. et al. Understanding of PE Headers and Analyzing PE File. *PalArch's Journal of Archaeology of Egypt / Egyptology*

- [online]. November 2020, zv. 17, č. 7, s. 8611–8620, [cit. 2022-03-07]. ISSN 1567-214x. Dostupné z: <https://www.archives.palarch.nl/index.php/jae/article/view/3670>.
- [9] DJEBBAR, F., AYAD, B., HAMAM, H. et al. A view on latest audio steganography techniques. In: *2011 International Conference on Innovations in Information Technology* [online]. Abu Dhabi, United Arab Emirates: IEEE, Apríl 2011, s. 409–414 [cit. 2022-01-29]. ISBN 978-1-4577-0314-0. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/5893859>.
- [10] DJEBBAR, F., AYAD, B., MERAİM, K. A. et al. Comparative study of digital audio steganography techniques. *EURASIP Journal on Audio, Speech, and Music Processing* [online]. Springer. Október 2012, zv. 25, č. 1, s. 1–16, [cit. 2022-01-29]. DOI: 10.1186/1687-4722-2012-25. ISSN 1687-4722. Dostupné z: <https://asmp-urasipjournals.springeropen.com/articles/10.1186/1687-4722-2012-25/>.
- [11] DOUGLAS, M., BAILEY, K., LEENEY, M. et al. An overview of steganography techniques applied to the protection of biometric data. *Multimedia Tools and Applications* [online]. CrossMark. Júl 2018, zv. 77, č. 13, s. 17333–17373, [cit. 2022-01-26]. DOI: 10.1007/s11042-017-5308-3. ISSN 1573-7721. Dostupné z: <https://link.springer.com/article/10.1007%2Fs11042-017-5308-3>.
- [12] EVSUTIN, O., MELMAN, A. a MESHCHERYAKOV, R. Digital Steganography and Watermarking for Digital Images: A Review of Current Research Directions. *IEEE Access* [online]. IEEE. September 2020, zv. 8, s. 166589–166611, [cit. 2022-01-06]. DOI: 10.1109/ACCESS.2020.3022779. ISSN 2169-3536. Dostupné z: <https://ieeexplore.ieee.org/document/9187785>.
- [13] FEBRYAN, A., PURBOYO, T. W. a SAPUTRA, R. E. Steganography Methods on Text, Audio, Image and Video: A Survey. *International Journal of Applied Engineering Research* [online]. Research India Publications. Január 2017, zv. 12, č. 21, s. 10485–10490, [cit. 2022-01-28]. ISSN 0973-4562. Dostupné z: <https://www.semanticscholar.org/paper/Steganography-Methods-on-Text-%2C-Audio-%2C-Image-and-%3A-Febryan-Purboyo/c74a460cbf3263b3081bd56810e4968bccea12a5>.
- [14] GORI, G. Tattooing as a vehicle for secret messages in ancient Greece. *Tattoo Life* [online]. November 2021 [cit. 2022-01-22]. Dostupné z: <https://www.tattoolife.com/tattooing-as-a-vehicle-for-secret-messages-in-ancient-greece/>.
- [15] HAMID, N., YAHYA, A., AHMAD, R. B. et al. Image Steganography Techniques: An Overview. *International Journal of Computer Science and Security (IJCSS)* [online]. Citeseer. 2012, zv. 6, č. 3, s. 168–187, [cit. 2022-01-28]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.735.5356&rep=rep1&type=pdf>.
- [16] HOPPER, N., AHN, L. von a LANGFORD, J. Provably Secure Steganography. *IEEE Transactions on Computers* [online]. IEEE. Máj 2009, zv. 58, č. 5, s. 662–676, [cit. 2022-01-24]. DOI: 10.1109/TC.2008.199. ISSN 1557-9956. Dostupné z: <https://ieeexplore-ieee-org.ezproxy.lib.vutbr.cz/document/4663056>.
- [17] HUSSAIN, M. a HUSSAIN, M. A survey of image steganography techniques. *International Journal of Advanced Science and Technology* [online]. Islamabad, Pakistan: Citeseer. Máj 2013, zv. 54, s. 113–124, [cit. 2022-01-29]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.364.3275>.

- [18] JOHNSON, N. F. a JAJODIA, S. Exploring steganography: Seeing the unseen. *Computer* [online]. IEEE. Február 1998, zv. 31, č. 2, s. 26–34, [cit. 2022-01-22]. DOI: 10.1109/MC.1998.4655281. ISSN 1558-0814. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/4655281>.
- [19] JOHRI, P., MISHRA, A., DAS, S. et al. Survey on Steganography Methods (Text, Image, Audio, Video, Protocol and Network Steganography). In: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* [online]. New Delhi, India: IEEE, Október 2016, s. 2906–2909 [cit. 2022-01-28]. ISBN 978-9-3805-4421-2. Dostupné z: <https://ieeexplore.ieee.org/document/7724795>.
- [20] KE, Y., LIU, J., ZHANG, M.-Q. et al. Steganography Security: Principle and Practice. *IEEE Access* [online]. IEEE. November 2018, zv. 6, s. 73009–73022, [cit. 2022-01-25]. DOI: 10.1109/ACCESS.2018.2881680. ISSN 2169-3536. Dostupné z: <https://ieeexplore.ieee.org/document/8537887>.
- [21] KOUR, J. a VERMA, D. Steganography Techniques – A Review Paper. *International Journal of Emerging Research in Management and Technology* [online]. Academia. Máj 2014, zv. 3, č. 5, s. 132–135, [cit. 2022-01-26]. ISSN 2278-9359. Dostupné z: [https://www.academia.edu/40997885/Steganography\\_Techniques\\_A\\_Review\\_Paper?bulkDownload=thisPaper-topRelated-sameAuthor-citingThis-citedByThis-secondOrderCitations&from=cover\\_page](https://www.academia.edu/40997885/Steganography_Techniques_A_Review_Paper?bulkDownload=thisPaper-topRelated-sameAuthor-citingThis-citedByThis-secondOrderCitations&from=cover_page).
- [22] KOWALCZYK, K. Portable Executable File Format. *Kowalczyk's Blog* [online]. Júl 2018 [cit. 2022-03-07]. Dostupné z: <https://blog.kowalczyk.info/articles/pefileformat.html>. Path: Home; File formats; pe format, pefile; Portable Executable File Format.
- [23] KRISHNAN, R. B., THANDRA, P. K. a BABA, M. S. An Overview of Text Steganography. In: *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)* [online]. Chennai, India: IEEE, Marec 2017, s. 1–6 [cit. 2022-01-29]. ISBN 978-1-5090-4740-6. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/8085643>.
- [24] MARELLA, K. S. Steganography — ‘The Dark cousin’ of cryptography. *Techiepedia: WritingTech* [online]. Január 2021 [cit. 2022-01-22]. Dostupné z: <https://medium.com/techiepedia/steganography-the-dark-cousin-of-cryptography-21d96a594068>. Path: Home; CYBERSECURITY; Steganography — ‘The Dark cousin’ of cryptography.
- [25] MIHARA, T. Misdirection steganography. *Soft Computing* [online]. Springer. November 2020, zv. 24, č. 21, s. 16005–16010, [cit. 2022-01-27]. DOI: 10.1007/s00500-020-05345-1. ISSN 1433-7479. Dostupné z: <https://link.springer.com/article/10.1007/s00500-020-05345-1>.
- [26] MSTAFA, R. J., ELLEITHY, K. M. a ABDELFAH, E. Video Steganography Techniques: Taxonomy, Challenges, and Future Directions. In: *2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT)* [online]. Farmingdale, NY, USA: IEEE, Máj 2017, s. 1–6 [cit. 2022-01-29]. ISBN 978-1-5386-3887-3. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/8001965>.

- [27] NISI, D., GRAZIANO, M., FRATANTONIO, Y. et al. Lost in the Loader: The Many Faces of the Windows PE File Format. In: ACM, ed. *24th International Symposium on Research in Attacks, Intrusions and Defenses* [online]. New York, NY, USA: Association for Computing Machinery, Október 2021, s. 177–192 [cit. 2022-03-06]. ISBN 9781450390583. Dostupné z: <https://dl.acm.org/doi/10.1145/3471621.3471848>.
- [28] NOSRATI, M., KARIMI, R. a HARIRI, M. An introduction to steganography methods. *World Applied Programming* [online]. Citeseer. August 2011, zv. 1, č. 3, s. 191–195, [cit. 2022-01-28]. ISSN 2222-2510. Dostupné z: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.208.5195&rep=rep1&type=pdf>.
- [29] RAKHI, S. G. A Review on Steganography Methods. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* [online]. Citeseer. Október 2013, zv. 2, č. 10, s. 4635–4638, [cit. 2022-01-28]. ISSN 2278–8875. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1038.2695&rep=rep1&type=pdf>.
- [30] SADEK, M. M., KHALIFA, A. S. a MOSTAFA, M. G. M. Video steganography: a comprehensive review. *Multimedia Tools and Applications* [online]. Springer. September 2015, zv. 74, č. 17, s. 7063–7094, [cit. 2022-01-28]. DOI: 10.1007/s11042-014-1952-z. ISSN 1573-7721. Dostupné z: <https://link.springer.com/article/10.1007/s11042-014-1952-z>.
- [31] SHARMA, S., GUPTA, A., TRIVEDI, M. C. et al. Analysis of Different Text Steganography Techniques: A Survey. In: *2016 Second International Conference on Computational Intelligence Communication Technology (CICT)* [online]. Ghaziabad, India: IEEE, Február 2016, s. 130–133 [cit. 2022-01-29]. ISBN 978-1-5090-0210-8. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/7546588>.
- [32] SHIH, F. Y. *Digital Watermarking and Steganography: Fundamentals and Techniques* [online]. 2. vyd. Boca Raton: CRC Press, apríl 2017 [cit. 2022-01-08]. 292 s. ISBN 9781315121109. Dostupné z: <https://www.taylorfrancis.com/books/mono/10.1201/9781315121109/digital-watermarking-steganography-fundamentals-techniques-frank-shih>.
- [33] SHIRALI SHAHREZA, M. H. a SHIRALI SHAHREZA, M. A New Synonym Text Steganography. In: *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing* [online]. Harbin, China: IEEE, August 2008, s. 1524–1526 [cit. 2022-01-29]. ISBN 978-0-7695-3278-3. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/4604331>.
- [34] SHIRALI SHAHREZA, M. Text Steganography by Changing Words Spelling. In: *2008 10th International Conference on Advanced Communication Technology* [online]. Gangwon, Korea (South): IEEE, Február 2008, sv. 3, s. 1912–1913 [cit. 2022-01-29]. ISBN 978-89-5519-136-3. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/4494159>.
- [35] SHIRALI SHAHREZA, M. a SHIRALI SHAHREZA, M. H. Text Steganography in SMS. In: *2007 International Conference on Convergence Information Technology (ICCIT 2007)* [online]. Gwangju, Korea (South): IEEE, November 2007, s. 2260–2265 [cit. 2022-01-29]. ISBN 0-7695-3038-9. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/4420590>.



- [36] TAHA, M. S., RAHIM, M. S. M., LAFTA, S. A. et al. Combination of Steganography and Cryptography: A short Survey. *IOP Conference Series: Materials Science and Engineering* [online]. IOP Publishing. Máj 2019, zv. 518, č. 5, s. 052003–052016, [cit. 2022-01-24]. DOI: 10.1088/1757-899x/518/5/052003. ISSN 1757-8981. Dostupné z: <https://doi.org/10.1088/1757-899x/518/5/052003>.
- [37] TAYEL, M., GAMAL, A. a SHAWKY, H. A proposed implementation method of an audio steganography technique. In: *2016 18th International Conference on Advanced Communication Technology (ICACT)* [online]. PyeongChang, Korea (South): IEEE, Február 2016, s. 180–184 [cit. 2022-01-29]. ISBN 978-8-9968-6506-3. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/7423320>.
- [38] TOOL INTERFACE STANDARD (TIS). *Executable and Linking Format (ELF)* [online]. Specification. Tool Interface Standard (TIS), máj 1995 [cit. 2022-03-05]. Dostupné z: <https://refspecs.linuxfoundation.org/elf/elf.pdf>.
- [39] WAHAB, O. F. A., KHALAF, A. A. M., HUSSEIN, A. I. et al. Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques. *IEEE Access* [online]. IEEE. Február 2021, zv. 9, s. 31805–31815, [cit. 2022-01-25]. DOI: 10.1109/ACCESS.2021.3060317. ISSN 2169-3536. Dostupné z: <https://ieeexplore.ieee.org/document/9356603>.
- [40] WESTFELD, A. a PFITZMANN, A. Attacks on Steganographic Systems: Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and STools – and Some Lessons Learned. In: PFITZMANN, A., ed. *Information Hiding* [online]. Berlin, Heidelberg: Springer, 2000, sv. 1768, s. 61–76 [cit. 2022-01-19]. Lecture Notes in Computer Science. ISBN 978-3-540-46514-0. Dostupné z: [https://link.springer.com/chapter/10.1007/10719724\\_5](https://link.springer.com/chapter/10.1007/10719724_5).
- [41] YAHYA, A., HAMID, N., AHMAD, R. B. et al. Steganography Techniques. In: YAHYA, A., ed. *Steganography Techniques for Digital Images* [online]. 1. vyd. Palapye, Botswana: Springer, Cham, Switzerland, 2019, kap. 2, s. 9–42 [cit. 2022-01-28]. ISBN 978-3-319-78597-4. Dostupné z: <https://link.springer.com/book/10.1007/978-3-319-78597-4>.

## Príloha A

# Špecifikácia metód jednotlivých digitálnych steganografií

Obsahom tejto prílohy sú štyri podkapitoly, ktoré popisujú základné skupiny metód jednotlivých steganografií. Navyše, pri každej skupine je uvedených pár základných steganografických techník reprezentujúcich tieto skupiny. Ide o steganografiu textovú, obrazovú, zvukovú a videosteganografiu, ktorým sa venujú podkapitoly [A.1](#), [A.2](#), [A.3](#) a [A.4](#) v tomto poradí.

### A.1 Textová steganografia

Táto podkapitola je doplnením podkapitoly [2.2](#) z hlavného textu práce. Jej obsahom je klasifikácia metód textovej steganografie, ktorá je nasledujúca:

#### A.1.1 Metódy založené na formáte

Tieto metódy používajú a menia formátovanie krycieho objektu (textu), čím skrývajú tajnú informáciu. Keďže ide len o formát textu, väčšina týchto metód nijako nemení samotný text krycieho objektu. Výnimkou môže byť napr. úmyselný preklep v rámci textu. Metódy tejto kategórie majú za cieľ byť spoľahlivo dekódovateľné, avšak pre čitateľa nerozoznateľné. Nasledujúce techniky sú len príkladmi tejto kategórie metód a je možné ich používať samostatne alebo aj spoločne: [\[5\]](#) [\[28\]](#) [\[19\]](#)

#### Metóda otvorených medzier (angl. *Open Space Method*)

Ide o pridávanie bielych znakov, konkrétne medzier, kde jedna medzera symbolizuje bit 0 a dve medzery bit 1. Tieto medzery sa môžu vyskytovať medzi slovami, vetami alebo odsekmi, alebo na konci riadkov. Aj keď je kapacita krycieho objektu výrazne obmedzená, metódu je možné použiť na akýkoľvek textový súbor, pričom odhalenie tajnej správy je veľmi obtiažne. Nevýhodou je, že niektoré textové editory môžu automaticky odstrániť nadbytočné medzery pri formátovaní, čím dôjde k strate ukrytých informácií. Taktiež veľkosť textového súboru sa vkladáním medzier zväčšuje. Napriek tomu má metóda využitie pri ukrytí informácií v rámci web stránky, pretože nadbytočné medzery v HTML dokumente neovplyvňujú zobrazenie výsledného dokumentu. [\[5\]](#) [\[31\]](#) [\[23\]](#)

### Kódovanie s posunom slov (angl. *Word-Shift Coding*)

Úprava dokumentu horizontálnym posunutím slov v riadkoch zabezpečí jedinečné zakódovanie. Slová sa delia do skupín po troch v každom riadku, pričom krajné slová zostávajú bez narušenia. Stredné slovo sa posunie smerom doľava na zakódovanie bitu 0 alebo doprava na zakódovanie bitu 1. Riadky v rámci dokumentu sú zarovnané, čo zníži pravdepodobnosť odhalenia tejto metódy. Techniku je možné použiť na formátovaný súbor alebo bitmapu textového dokumentu, avšak použiteľná je len pre dokumenty s premenlivými medzerami medzi susednými slovami. Premennivé medzery sú často používané kvôli rozdeleniu medzier pri zarovnaní textu. Pre extrakciu skrytej informácie je nutný pôvodný nezakódovaný dokument. Túto techniku znázorňuje obrázok A.1. [28] [23] [34]

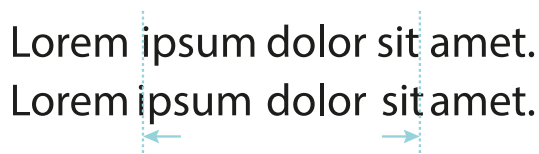


Diagram illustrating Word-Shift Coding. It shows two lines of text: "Lorem ipsum dolor sit amet." The top line is the original text with spaces. The bottom line shows the text after encoding, where the middle word "ipsum" is shifted left and "dolor" is shifted right, indicated by blue arrows. Vertical dashed lines align the words between the two lines.

Obr. A.1: Ukážka kódovania s posunom slov – vrchný riadok znázorňuje medzery medzi slovami pred zakódovaním informácie, pričom spodný po zakódovaní. (Prevzaté a upravené z [23])

### Kódovanie s posunom riadkov (angl. *Line-Shift Coding*)

Jedinečné zakódovanie vytvorí úprava dokumentu vertikálnym posunutím riadkov textu. Ako pri predchádzajúcej metóde, aj tu existujú skupiny pozostávajúce z troch po sebe idúcich riadkov. Vždy krajné riadky zostávajú nenarušené, pričom stredný sa posunie smerom nahor, na zakódovanie bitu 0, alebo nadol, na zakódovanie bitu 1. Krajné riadky tak vytvárajú akési „čiary“, ktoré sa pri dekódovaní používajú na kontrolu, či bol stredný riadok posunutý alebo nie. Opäť je možné použiť bitmapu alebo formátovaný text. Existujú však prípady, kedy je možné dekódovanie úspešne uskutočniť bez prítomnosti pôvodného nezakódovaného dokumentu, pretože za bežných okolností je v celom dokumente rovnaké riadkovanie. Túto techniku znázorňuje obrázok A.2. [28] [23] [35]

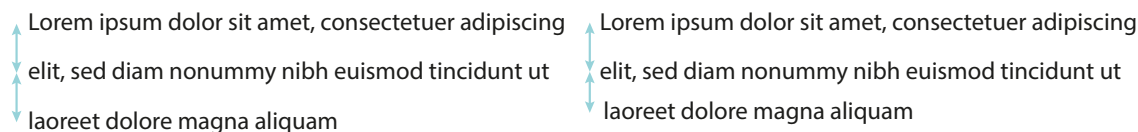


Diagram illustrating Line-Shift Coding. It shows two columns of text. The left column is the original text with line breaks. The right column shows the text after encoding, where the middle line is shifted up or down, indicated by blue arrows. Vertical dashed lines align the lines between the two columns.

Obr. A.2: Ukážka kódovania s posunom riadkov – obrázok vľavo znázorňuje riadkovanie pred zakódovaním informácie, pričom ten vpravo po zakódovaní. (Prevzaté a upravené z [23])

### Kódovanie vlastností (angl. *Feature Coding*)

Opäť je možné kódovanie formátovaného súboru alebo bitmapy textového súboru. Cieľom je skúmanie vybraných textových prvkov, ktoré sa pozmenia alebo nepozmenia v závis-

losti od kódovanej informácie. Dekódovanie vyžaduje pôvodný dokument alebo presnejšiu špecifikáciu zmeny v pixeloch daného prvku. [28] [35]

Príkladom takejto metódy je *kódovanie zvislých koncových čiar písmen* (napr. b, d, h, k, ...) ich predĺžením o jeden či viac pixelov, pre kódovanie bitu 0, a ich neporušením, pre kódovanie bitu 1. [23]

Iným použiteľným príkladom metódy, ktorá kóduje informáciu pomocou nejakej vlastnosti textu je tzv. *pohyb bodky v znakoch*. Malé písmená anglickej abecedy „i“ a „j“, rovnako ako arabské či perzské znaky abecedy, majú bodky. Bodky týchto znakov je možné posunúť nahor, pre zakódovanie bitu 0, alebo ich nechať nenarušené, pre zakódovanie bitu 1. Techniku znázorňuje obrázok A.3. [23]



Obr. A.3: Ukážka kódovania s vybranou vlastnosťou textu – písmeno vľavo má bodku neporušenú (bit 1), pričom písmeno vpravo ju má posunutú vyššie (bit 0). (Prevzaté a upravené z [23])

### A.1.2 Náhodné a štatistické metódy

Tieto metódy generujú krycí text podľa štatistických vlastností daného jazyka a vkladajú tajnú informáciu tak, že sa vyskytuje v náhodnom poradí znakov. Toto poradie sa musí zdať náhodné aj pre prípadného útočníka. Metódy využívajú vzorové gramatiky určitého prirodzeného jazyka: [29] [19]

#### Pravdepodobnostná bezkontextová gramatika

Je to bežne používaný jazykový model, kde je priradená pravdepodobnosť pre každé transformačné pravidlo bezkontextovej gramatiky. Tento model je možné použiť na generovanie slovných sekvencií tak, že sa začne od koreňového uzla a rekurzívne sa aplikujú náhodne zvolené pravidlá. Vety sú zostavené podľa tajnej informácie, ktorá sa v nej má skrývať. Kvalita vygenerovaného stego-objektu priamo závisí od kvality použitých gramatík. [5]

#### Generovanie slov s rovnakými štatistickými vlastnosťami

Metóda generuje slová s rovnakými štatistickými vlastnosťami, ako je dĺžka slova či frekvencia písmen. Vygenerované slová často nemajú žiadnu lexikálnu hodnotu. [5]

### A.1.3 Lingvistické metódy

Ide o umenie využívať prirodzený jazyk na ukrytie tajných informácií. Na úpravu textu sa využívajú jeho jazykové vlastnosti. Ide teda o kombináciu syntaktiky a sémantiky jazyka: [19] [29]

## Syntaktická metóda

Pri tejto metóde sa umiestňujú interpunkčné znamienka (čiarka, bodka, ...) na správne miesta tak, aby bolo možné ukryť tajnú informáciu. Metóda vyžaduje správnu identifikáciu miest na umiestnenie týchto znamienok. Je však možné takto ukryť len malé množstvo informácií. [33] [34]

## Sémantická metóda

Táto metóda skrýva informácie zámenou určitých slov za ich synonymá. Takáto substitúcia môže ukryť jeden alebo viac bitov tajnej informácie. Je tiež odolná voči prepisovaniu, čím do istej miery chráni tieto informácie. Niekedy však dôjde k zmene významu samotného textu. [31] [34] [33]

## A.2 Obrazová steganografia

Doplnením podkapitoly 2.3 z hlavného textu práce sa zaoberá táto podkapitola. Jej obsahom je klasifikácia metód obrazovej steganografie, ktorá je nasledujúca:

### A.2.1 Metódy priestorovej domény

*Metódy priestorovej domény*, nazývané aj *substitučné techniky*, sú skupinou relatívne jednoduchých techník, ktoré využívajú slabé stránky ľudského zraku. Preto je možné skryť informáciu do *najmenej významných bitov* (ďalej len LSB<sup>1</sup>) krycieho obrázka, čo predstavuje rovnomennú techniku. V rámci obrázka je možné brať LSB za náhodný šum, ktorý v ňom žiadne zmeny nepredstavuje. [15] [41] [17]

Algoritmus LSB môže mať pri vkladaní informácie do obrázka dve schémy:

- **Sekvenčná** – každý LSB obrázka je postupne nahradený bitmi informácie.
- **Rozptýlená** – bity informácie sú náhodne rozptýlené do LSB obrázka pomocou náhodnej sekvencie, ktorá toto vkladanie reguluje.

Steganografické nástroje založené na metódach LSB sú rôzne. Niektoré modifikujú LSB každého pixelu, pričom iné len vo vybraných oblastiach obrázka. Výhodou techník LSB je vysoká kapacita a nízka degradácia krycieho obrázka. Veľkou nevýhodou je, že nie sú robustné voči stratovej kompresii, ktorá je – ako už bolo zmienené – pri digitálnych obrázkoch veľmi častá. [15] [41] [17]

Prehľad niektorých použiteľných variantov LSB techník (okrem samotnej LSB metódy) [17] [29]:

- *Rozdiel hodnôt pixelov* (angl. *Pixel Value Differencing – PVD*)
- *Vkladanie dát založené na hranách* (angl. *Edges Based Data Embedding*)
- *Vkladanie do náhodných pixelov* (angl. *Random Pixel Embedding*)
- *Metóda založená na textúre* (angl. *Texture Based Method*)

---

<sup>1</sup>Najmenej významný bit (angl. *Least Significant Bit – LSB*)

### A.2.2 Metódy transformačnej domény

V súčasnosti takmer všetky robustné steganografické algoritmy fungujú na vkladanie informácií v rámci *transformačnej domény*. Je to z dôvodu, že vkladanie dát do frekvenčnej oblasti signálu je dostatočne odolné v porovnaní s metódami časovej domény. Preto sa občas v literatúre v súvislosti s touto skupinou metód objavuje termín *metódy frekvenčnej domény*. Techniky transformačnej domény sú výhodnejšie v porovnaní s technikami LSB, lebo skrývanie dát je zamerané na oblasti obrázka, ktoré sú menej vystavené kompresii či inému spracovaniu. Niektoré metódy tejto kategórie sú tiež nezávislé od formátu obrázka, čo implikuje odolnosť voči bezstratovej aj stratovej kompresii obrázka. Techniky transformačnej domény sa vo všeobecnosti delia na: [17] [29]

1. *Diskrétna Fourierová transformácia* (angl. *Discrete Fourier Transform – DFT*)
2. *Diskrétna kosínusová transformácia* (angl. *Discrete Cosine Transform – DCT*)
3. *Diskrétna vlnová transformácia* (angl. *Discrete Wavelet Transform – DWT*)

Najbežnejším obrazovým formátom používaným na internete je formát JPEG. Vo väčšine JPEG steganografických systémov sa informácie vkladajú do nenulových koeficientov DCT. Nasledujúce známe JPEG steganografické techniky sú toho dôkazom. [15] [41]

### F5

Algoritmus vkladá informácie do absolútnej hodnoty nenulových koeficientov DCT znížením ich hodnoty o jedna namiesto náhrady LSB koeficientov DCT za bity informácie. Táto technika je absolútne imúnna voči vizuálnym útokom na stego-obrázok. Za účelom zníženia šumu zabudovaného do signálu sa používa maticové kódovanie. Ide o jednu z najpopulárnejších schém vkladania v doméne DCT. [15] [29]

### OutGuess

Existujú dve verzie tejto techniky. Prvá, *OutGuess-0.13b*, je primárna verzia zobrazujúca štatistickú analýzu. Druhá, *OutGuess-0.2*, umožňuje ochranu voči štatistickým útokom. Implicitne sa technikou OutGuess myslí práve druhá verzia. Proces vkladania OutGuess pozostáva z náhodného vloženia bitov informácie do LSB koeficientov DCT, avšak s vyhýbaním sa nulám a jednotkám. Týmto vzniká komplexný histogram DCT pre stego-obrázok, ktorý je ekvivalentný s histogramom DCT pre ten pôvodný. Technika však imúnna voči vizuálnym útokom nie je. [15] [41]

### MB

Technika založená na modeli *MB* môže byť definovaná ako všeobecný rámec na vykonávanie steganografie aj steganalýzy jednoduchým použitím štatistického modelu krycieho obrázku. Schéma MB má v prípade JPEG obrázkov vysokú kapacitu a je bezpečná voči štatistickým útokom prvého stupňa. [15] [41]

### YASS

Alternatívny prístup k vkladaniu do JPEG obrázkov má schéma YASS<sup>2</sup>. Vstupný obrázok sa rozdelí do blokov s bezpečne veľkou veľkosťou – *B-bloky*. V ďalšom kroku sa náhodne vyberie

<sup>2</sup>Ďalšia steganografická schéma (angl. *Yet Another Steganographic Scheme – YASS*)

v každom B-bloku podblok o rozmere  $8 \times 8$  – *H-blok*. Pomocou kódov na opravu chýb sa šifrovaná informácia vloží do koeficientov DCT v H-bloku. Nakoniec sa obrázok komprimuje a je distribuovaný vo formáte JPEG po inverzii koeficientov DCT na H-blokoch. [15] [41]

### Technika vlnovej transformácie

DWT konvertuje informácie o priestorovej doméne na informácie o frekvenčnej doméne. Vlny sa používajú v obraze, pretože DWT jasne rozdeľuje vysokofrekvenčné a nízkofrekvenčné informácie pixel po pixeli. Metóda DWT je uprednostňovaná pred metódou DCT vďaka rozlíšeniu, ktoré DWT poskytuje obrazu na rôznych úrovniach. Vlny sú matematické funkcie, ktoré rozdeľujú údaje na frekvenčné zložky, vďaka čomu sú ideálne na kompresiu obrazu. [29] [15] [41]

### A.2.3 Metódy rozprestretého spektra

Schémy *metód rozprestretého spektra* spĺňajú maximálne požiadavky schém na skrytie informácií, najmä pokiaľ ide o štatistické hrozby. Z tohto dôvodu sú skryté údaje rozptýlené po celom obraze bez zmeny štatistických vlastností. Celkovo možno metódy rozprestretého spektra použiť vo väčšine steganografických aplikácií, napriek tomu, že sú charakteristické tým, že sú vysoko matematickým a zložitým prístupom. Tieto metódy sa v steganografii opierajú buď o krycí obrázok ako šum, alebo sa pokúšajú pridať ku kryciemu obrázku pseudošum. [41] [15]

### Krycí obrázok ako šum

Takýto steganografický systém zaobchádza s krycím obrázkom ako so šumom a môže tomuto obrázku priradiť jeden bit. Pre prípad prenosu viac bitov sa krycí obrázok rozdelí do krycích podobrázkov. Potom ide o steganografiu s rozprestretým spektrom priamej sekvencie. Keď sa krycie podobrázky skladajú zo samostatných bodov rozmiestnených po krycom obraze, táto technika sa označuje ako steganografia s rozprestretým spektrom s preskakovaním frekvencie. Obe techniky sú odolné voči jemnej kompresii JPEG. [15]

### Pseudošum

Technika ukazuje, že skrytá informácia je rozptýlená po celom krycom obraze, a preto je ju ťažké odhaliť. Príkladom tejto techniky je *steganografia obrazu s rozprestretým spektrom*. Jej proces začína ukrytím informácie v šume a následne sa skombinuje s krycím obrázkom, čím sa dostane do stego-obrázku. Keďže sila vloženého signálu je oveľa nižšia ako sila krycieho obrázka, stego-obrázok sa stáva nepostrehnuteľným nielen pre ľudský zrak, ale aj pre steganalýzu bez prístupu k pôvodnému obraze. [15]

V rámci obrazovej steganografie sa zistilo, že vysoké frekvencie zvyčajne pomáhajú pri neviditeľnosti skrytých informácií, no zároveň nie sú veľmi robustné. Naopak, nízke frekvencie podmieňujú lepšiu robustnosť na úkor viditeľnosti, čo znamená, že sú nepoužiteľné. Túto konfliktnú situáciu zosúlaďuje technika rozprestretého spektra tým, že umožňuje vložiť nízkoenergetický signál do každého z frekvenčných pásiem. Techniky rozprestretého spektra je tiež možné kombinovať s transformačnými technikami, aby sa zvýšila kapacita. [15]



#### A.2.4 Štatistické metódy

Ide o techniky, ktoré upravujú štatistické vlastnosti obrázka v procese vkladania. Štatistické steganografické techniky označujú použitie existencie jednobitového steganografického systému, v ktorom je jeden bit dát vložený do obrázka. Proces vloženia pozostáva z jednoduchéj a malej úpravy obrázka tak, aby nastala významná zmena v štatistických charakteristikách – vtedy ide o zakódovanie bitu 1. Ak obrázok zostane nezmenený, znamená to zakódovanie bitu 0. Je tiež možné zakódovať viacbitovú informáciu, kedy sa obrázok rozdelí na samostatné bloky (podobrázky), pričom každý predstavuje jeden bit informácie. [15] [41]

Inou štatistickou metódou je použitie vodoznaku, ktorý poskytuje základ pre štatistickú funkciu. Keďže sa vodoznaky považujú za ťažko rozpoznateľné a ťažko odstrániteľné, pričom sa dajú ľahko obnoviť – za predpokladu znalosti kľúča. Bloky obrázka, ktoré majú zakódovať bit 1 sa označia vodoznakom. Bloky, ktoré nie sú označené vodoznakom, kódujú bit 0. Niekedy sa v literatúre táto metóda radí do skupiny *maskovacích a filtrovacích metód* (angl. *Masking and Filtering Methods*), ktorých výhodou oproti LSB technikám je, že maskujú vkladajú informáciu vo viditeľných častiach obrázku a nie na úrovni šumu. Tým sa stavajú odolnými voči stratovej kompresii (napr. JPEG) a rôznym spracovaniám obrázku. [41] [28]

Štatistické metódy sú náchylné na útoky orezania, rotácie a zmeny mierky obrázka, a tiež na útoky, na ktoré je náchylný samotný vodoznak. Účinnou obranou voči takýmto útokom môže byť koncept, ktorý vytvára bloky v obrázku na základe jeho obsahu (napr. bloky predstavujú tváre na obrázku) a používa kódovanie na opravu chýb v rámci vkladanej informácie. Tento koncept môže nadobudnúť robustnosť rovnakú, akú má vodoznak. Avšak tieto metódy nie sú odolné voči steganalýze, ktorá meria štatistické vlastnosti obrázku, preto sú štatistické metódy v praxi menej použiteľné v porovnaní s inými metódami. [41] [15]

#### A.2.5 Techniky skreslenia

*Techniky skreslenia* vyžadujú znalosť pôvodného krycieho obrázku kvôli procesu dekodovania, kde dekodér funguje na kontrole rozdielov medzi pôvodným obrázkom a skresleným obrázkom, čím sa dekoduje vložená informácia. Vloženie informácie teda znamená skreslenie obrázku. Skreslenie obrázku je vykonané na základe kódovanej informácie. Tá sa kóduje v náhodne vybraných pixeloch. Ak sa stego-obrázok líši od pôvodného v danom pixeli, znamená to, že tento pixel kóduje bit 1, inak kóduje bit 0. Výhodou techniky je možnosť upraviť pixeli tak, aby sa zachovali štatistické vlastnosti obrázka, čím sa technika stáva odolnou voči štatistickým útokom. Vďaka tomu sa techniky skreslenia líšia od metód LSB. [15] [17] [41]

Nevýhodou, ktorá komplikuje použitie tejto techniky, je nutnosť odoslať aj pôvodný obrázok. Taktiež by sa jeden krycí obrázok nemal použiť viackrát – platí pre každú steganografickú techniku. V niektorých prípadoch, keď je informácia vložená kódovaním na opravu chýb, je táto technika odolná voči útokom pozmeňujúcim stego-obrázok, pretože vloženú informáciu je možné úplne obnoviť. [15] [17] [41]

### A.3 Zvuková steganografia

Podkapitoly 2.4 z hlavného textu práce dopĺňa táto podkapitola. Jej obsahom je klasifikácia metód zvukovej steganografie, ktorá je nasledujúca:

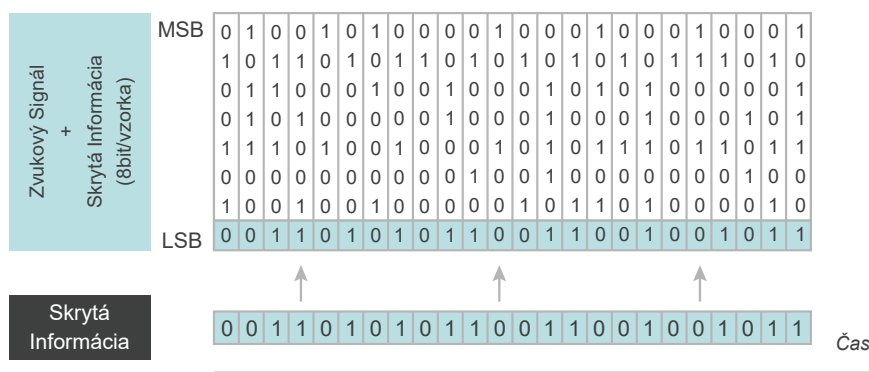


### A.3.1 Metódy časovej domény

Väčšina metód tejto kategórie využíva techniky LSB a jej variantov. Pre túto skupinu metód nie sú hlavnými znakmi robustnosť ani bezpečnosť, no technika LSB a jej varianty poskytujú jednoduchý spôsob ukrytia informácií. V súčasnosti bolo vyvinutých len niekoľko metód tejto kategórie, pričom nasledujúce uvedené techniky sú len niektorými z nich: [10]

#### Kódovanie najmenej významných bitov (angl. *LSB Coding*)

Jednou z prvých, najjednoduchších a bežne používaných techník pre zvukovú steganografiu je práve kódovanie LSB. Technika pozostáva z vloženia každého bitu informácie do bitu LSB krycieho zvukového signálu. Aj keď je táto metóda veľmi jednoduchá, nedokáže ochrániť skrytú informáciu ani pred malými úpravami, ktoré môžu vzniknúť za rôznych situácií (napr. konverzia formátu, ...). Technika LSB sa dá ľahko implementovať a kombinovať s inými efektívnejšími technikami ukrývania informácií. Jej ďalšou výhodou je vysoká kapacita na prenos mnohých typov digitálnych objektov, avšak dĺžka tajnej informácie by mala byť menšia než celkový počet vzoriek krycieho signálu. LSB taktiež využíva nedokonalosť ľudského sluchu, ktorý nevie rozpoznať malé odchýlky frekvencií zvukového signálu. Okrem toho je LSB veľmi rýchla a efektívna technika, pričom sa kvalita signálu neznižuje. Techniku LSB znázorňuje obrázok A.4. [37] [5]



Obr. A.4: Ukážka kódovania LSB (prevzaté a upravené z [10])

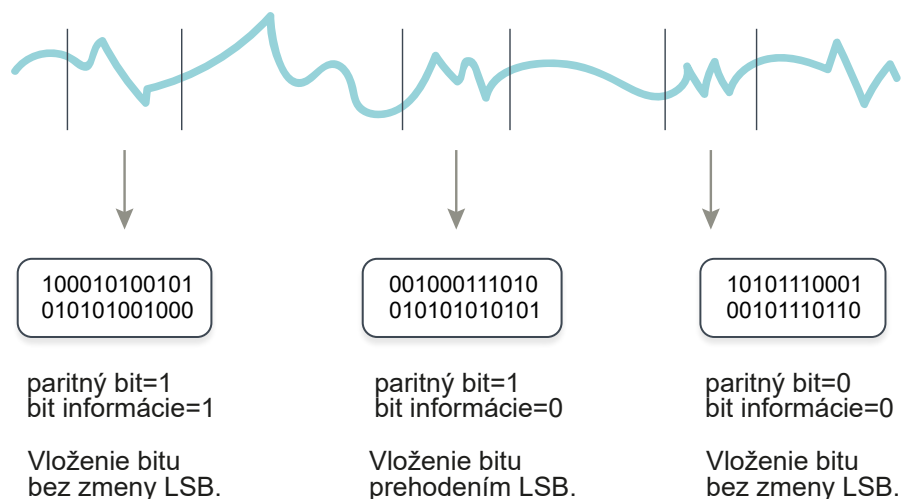
#### Skrytie ozveny (angl. *Echo Hiding*)

Tajné informácie sú vložené ako ozvena do krycieho zvukového signálu. Ozvena je rezonancia pridaná ku kryciemu signálu, čo spôsobí, že sa predíde problémom s aditívnym šumom. Na úspešné skrytie informácie je potrebné zmeniť počiatočnú amplitúdu, rýchlosť doznievania (poklesu) a uskutočniť posun (oneskorenie) od pôvodného signálu tak, aby sa ozvena stala počuteľnou a tým reprezentuje zakódovanú tajnú informáciu. Ozvenu nie je možné ľahko detegovať, pretože všetky tri parametre sú pod hranicou ľudského sluchu. Kvôli nízkej bezpečnosti a rýchlosti sa vo výskume týchto techník ďalej nepokračuje. [37] [28]

#### Paritné kódovanie (angl. *Parity Coding*)

Tieto techniky pracujú so skupinami vzoriek namiesto jednotlivých. Teda jednotlivé vzorky sa zoskupia a vypočíta sa ich parita. Na vkladanie bitov informácie po jednom sa kontroluje

paritný bit skupiny vzoriek. Ak sa tento bit zhoduje s bitom informácie, nedeje sa nič. Ak sa však tieto bity nezhodujú, zmení sa LSB ktorejkoľvek z jednotlivých vzoriek v rámci danej skupiny tak, aby sa paritný bit rovnal bitu informácie. Túto techniku znázorňuje obrázok A.5. [37] [13]



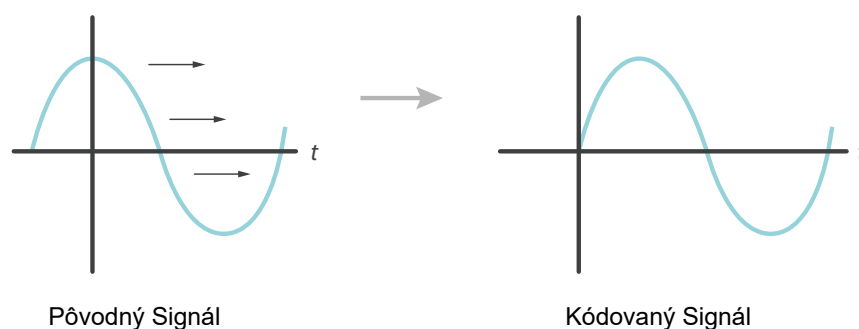
Obr. A.5: Ukážka paritného kódovania (prevzaté a upravené z [37])

### A.3.2 Metódy transformačnej domény

Táto skupina metód využíva tzv. „fenomén maskovacieho efektu“, kedy sa maskujú slabšie frekvencie blízko silnejších rezonančných. Tento fenomén využíva nedokonalosti ľudského sluchu a umožňuje tak veľmi efektívne skrytie informácií. Je známe, že skrytie informácie v transformačnej doméne, na rozdiel od časovej, poskytne lepšie výsledky z hľadiska pomeru signálu k šumu. V poslednom čase bolo vyvinutých veľa techník tejto kategórie, ktorým sa podarilo lepšie realizovať bezpečnosť a robustnosť. Preto sú do určitej miery skryté informácie odolné voči prevzorkovaniu, filtrácii alebo zosilneniu zvukového signálu. Na druhú stranu, pravdepodobne neprežijú hlučné prenosové prostredie alebo kompresiu. Následne sú uvedené len niektoré techniky z tejto kategórie metód: [10]

#### Fázové kódovanie (angl. *Phase Coding*)

Ľudský sluch nevie rozpoznať fázovú zmenu zvukového signálu tak, ako dokáže rozpoznať šum v signáli. Preto táto metóda túto skutočnosť využíva. Technika zakóduje bity tajnej informácie ako fázové posuny vo fázovom spektre digitálneho signálu. Tým sa dosiahne nepočuteľné kódovanie v zmysle pomeru signálu k vnímanému šumu, preto vzniká odolnosť voči steganalýze založenej na šume. Fázové kódovanie týmto rieši nevýhody metód zvukovej steganografie vyvolávajúcich šum. Technika je odolná voči skresleniu signálu, ale nevydrží dolnopriepustnú filtráciu. Techniku znázorňuje obrázok A.6. [5] [28] [37]



Obr. A.6: Ukážka posunu fázy zvukového signálu vpravo (prevzaté a upravené z [37])

### Kódovanie rozprestretého spektra (angl. *Spread Spectrum Coding*)

Pri základnej metóde sa náhodne rozložia bity tajnej informácie cez frekvenčné spektrum zvukového signálu. Táto metóda, na rozdiel od kódovania LSB, šíri tajnú informáciu pomocou kódu. Tento kód je nezávislý od skutočného krycieho signálu a je známy odosielateľovi aj príjemcovi. Táto metóda môže fungovať lepšie ako techniky kódovania LSB a fázy vďaka miernej rýchlosti prenosu dát v spojení s vysokou úrovňou odolnosti voči technikám steganalýzy. Avšak, podobne ako metóda kódovania LSB, aj táto metóda môže do zvukového signálu vniesť šum. Vytvára sa tak zraniteľnosť, ktorú je možné využívať pri steganalýze. [5] [37]

### Techniky vlnovej domény (angl. *Wavelet Domain Methods*)

Ide o zvukovú steganografiu založenú na diskretnej vlnovej transformácii. Informácie sa skrývajú v LSB koeficientov vlnovej transformácie audio signálu, pričom sa dosahuje vysokej kapacity až 200 kbps v 44,1 kHz zvukovom signáli. Pre zlepšenie nepostrehnuteľnosti vlozenej informácie je možné použiť prah počutia pri vkladaní informácií do celočíselných koeficientov. Skrytie informácií vo vlnovej doméne je síce rýchle, no extrakcia informácie príjemcom nemusí byť presná. [10] [9]

## A.4 Videosteganografia

Posledná podkapitola, ktorá dopĺňa podkapitolu 2.5 z hlavného textu práce. Jej obsahom je klasifikácia metód videosteganografie, ktorá je nasledujúca:

### A.4.1 Substitučné metódy

Techniky tejto kategórie metód sú založené na substitúcii a nahrádzajú redundantné informácie krycieho objektu za požadovanú tajnú správu. Ich výhodami sú vysoká kapacita pre vložené informácie a jednoduchosť implementácie v porovnaní s ostatnými. Nasledujúce techniky sú len príkladmi substitučných metód. [30]

### Technika najmenej významného bitu (angl. *LSB Technique*)

Technika dokáže skryť pomerne veľké množstvo bitov tajnej informácie nahradením niektorých najmenej významných bitov jednotlivých pixelov krycieho videa. Experimenty v

[30] dokazujú, že maximálny počet nahradených LSB bitov sú štyri. Je to z dôvodu začínajúceho vizuálneho skreslenia krycieho videa, ktoré je samozrejme závislé od použitých farieb vkladanej informácie (napr. obrazovej). Technikou LSB je inšpirovaná väčšina substitučných metód. Existuje niekoľko variantov metódy LSB od takých, ktorých vizuálne skreslenie optimalizované je, no nie sú dostatočne robustné alebo majú nízku kapacitu, až po také, ktoré značne modifikujú niektoré vlastnosti základnej techniky LSB. [30]

Napokon jednoduchá implementácia a nízka výpočtová náročnosť techniky LSB priťahla pozornosť a začala sa využívať na steganografiu v reálnom čase. Ide o ukrytie tajnej informácie v snímkach reklamných billboardov, kde každá snímka je rozdelená na malé bloky, v ktorých je ukrytá informácia. Na zabezpečenie je možné použiť tajný kľúč. Takúto techniku je možné použiť na vysielanie tajnej informácie na verejných miestach (parky, obchodné centrá, ...). Technika nepotrebuje žiaden úložný priestor, čo je veľmi výhodné pri skrývaní informácie v reálnom čase. Keďže nie je k dispozícii ani krycí objekt, nie je možné vykonávať jeho analýzu. Jediným obmedzením je, pochopiteľne, nutnosť pripojiť elektronické zariadenie implementujúce takúto techniku k zariadeniu vysielajúcemu v reálnom čase. Pre extrakciu informácie si príjemca musí zaznamenávať snímky obrazu. Ak zmešká nejakú snímku obsahujúcu informáciu, prichádza o ňu. [30]

Vo všeobecnosti je myšlienkou metódy LSB nahradenie najmenej významných bitov za iné, čo vedie k zhoršeniu kvality krycieho videa. Tento problém sa snažia riešiť nasledujúce metódy. [30]

### **Segmentácia zložitosti bitovej roviny (angl. *Bit Plane Complexity Segmentation – BPCS*)**

Táto metóda využíva slabé stránky ľudského zraku, ktorý nedokáže prijímať informácie o obraze v komplikovanom binárnom vzore. BPCS dokáže pracovať v priestorovej aj v transformačnej doméne. Myšlienkou metódy BPCS je rozdeliť snímku do bitových rovín. Bitová rovina sa môže považovať za výrez snímku, ktorý je tvorený všetkými bitmi špecifickej významovej pozície z každej binárnej číslice. Keď sú identifikované bitové roviny snímky, zmeria sa zložitosť každej oblasti tejto roviny. Oblasti sú rozdelené do dvoch typov: [30]

1. *informatívna oblasť a*
2. *oblasť podobná šumu*

Informatívna oblasť zostáva neporušená, no do oblasti pripomínajúcej šum sa vkladajú bity tajnej informácie, čo má za následok minimálne zníženie kvality videosnímk.

### **Trojcestný rozdiel hodnoty pixelov (angl. *Tri-way Pixel-Value Differencing – TPVD*)**

Ide o modifikáciu známej metódy *rozdiel hodnoty pixelov* (angl. *Pixel-Value Differencing – PVD*). Ukrytie tajnej informácie pomocou PVD je založené na rozdieli hodnôt dvoch susedných pixelov. Hodnoty týchto rozdielov sa delia do rozsahov, pričom sa každý skladá z dolnej hranice, hornej hranice a šírky rozsahu. Menší index rozsahu indikuje hladkú oblasť a vyšší ostrú oblasť. Väčšie množstvo informácie je možné vložiť do ostrej oblasti na rozdiel od tej hladkej. [30]

Pred ukrytím údajov sa krycia snímka rozdelí na neprekrývajúce sa bloky dvoch susedných pixelov. Následne sa určí hodnota rozdielu a rozsah. Potom sa vypočíta počet bitov informácie, ktoré sa môžu skryť, a to na základe indexu rozsahu. V tejto chvíli sa zistený

počet bitov z informácie extrahuje a ukryje. Ich príslušná desatinná hodnota sa ďalej použije na vytvorenie nového rozdielu, podľa ktorého sa upravujú hodnoty pixelov. Technika TPVD vkladá informáciu do všetkých vertikálnych, horizontálnych a diagonálnych okrajov, čo zvýši kapacitu krycieho snímku. [30]

#### A.4.2 Metódy transformačnej domény

Napriek rôznym modifikáciám, substitučné algoritmy neustále bojujú so slabou odolnosťou voči modifikácii krycieho objektu, ako sú kompresia, zmena formátu atď. *Metódy transformačnej domény* sú síce zložitejšie, no s vyššou robustnosťou a transparentnosťou vnímania kvality vzniknutých stego-objektov. Zásadou každej techniky tejto kategórie je transformácia krycieho videa do frekvenčnej domény s následným vložením tajnej informácie do niektorých alebo všetkých transformovaných koeficientov. Posledným krokom na záver je spätná transformácia zmenených koeficientov do pôvodného krycieho videa. Takéto transformácie je možné uskutočniť diskretnou Fourierovou transformáciou (DFT<sup>3</sup>), diskretnou kosínusovou transformáciou (DCT<sup>4</sup>) a diskretnou vlnovou transformáciou (DWT<sup>5</sup>). Vo videosteganografii sa častejšie používajú metódy založené na DCT a DWT. Metódy DFT sa ukázali ako neoptimálne, pretože zavádzajú veľké zaokrúhľovacie chyby. [30]

#### A.4.3 Adaptívne metódy

*Adaptívne metódy* sú novou technikou vkladania, hovorí sa im tiež *maskovacie metódy*. Jej myšlienkou je analýza štatistických znakov krycieho videosúboru pred vložením tajnej informácie. Výsledkom analýzy je identifikácia najvhodnejších oblastí pre vloženie informácie – *oblasti záujmu*. Okrem toho môže byť výstupom analýzy aj počet bitov informácie, ktoré sa majú skryť. Počet závisí od funkcie adaptívnej kapacity. V podstate je táto skupina metód len špeciálnym prípadom techník z ostatných kategórií tejto klasifikácie. V konečnom dôsledku pre dosiahnutie lepšej kvality stego-video je krycie video adaptívne upravené podľa niekoľkých kritérií. Svoj adaptívny variant má aj metóda LSB. [30]

#### A.4.4 Metódy založené na formáte

Ako z názvu vyplýva, tieto techniky sú metódy navrhnuté priamo pre konkrétny formát videosúboru. V súčasnosti je navrhnutých viacero použiteľných formátov, ktoré je takto možné použiť ako krycie videá. Jedným z najnovších štandardov kompresie videosúboru je H.264/AVC, ktorý poskytuje vysokú účinnosť kompresie a je vhodný pre rýchly prenos po sieti. Metódy založené na tomto formáte je viacero, pretože môžu efektívne využívať jeho štruktúru. [30]

Ďalším užitočným formátom je formát videosúborov Flash (prípona .FLV). Oblúbenosť formátu na internete je do značnej miery zapríčinená jeho jednoduchou štruktúrou a malou veľkosťou v porovnaní s inými. Príklad algoritmu pre tento formát je založený na myšlienke rovnomerného rozdelenia tajnej informácie medzi značky (angl. *tags*) videa v celom súbore, pričom jednotlivé časti informácie sú vložené za každú značku tak, aby sa skutočné značky videa a jeho zvuku neovplyvnili – nemodifikovali a ani nevynechali. Týmto zostáva kvalita videa úplne nezmenená a bez akéhokoľvek skreslenia. Výhodou je, že je možné vlo-

<sup>3</sup>Z angl. Discrete Fourier Transform – DFT

<sup>4</sup>Z angl. Discrete Cosine Transform – DCT

<sup>5</sup>Z angl. Discrete Wavelet Transform – DWT

žiť informáciu neobmedzenej veľkosti, s čím sa ale zvyšuje veľkosť krycieho videa. Navyše, algoritmus nie je robustný. [30]

#### A.4.5 Metódy generujúce krycí videosúbor

Všetky vyššie uvedené metódy využívajú určitý krycí objekt, na ktorom aplikujú steganografický algoritmus. Táto sekcie približuje metódy, ktoré syntetizujú objekt na to, aby ho mohli použiť ako krycí v rámci výmeny tajných informácií. V tomto prípade ide o myšlienku generovania dynamického krycieho videa. Tento proces si vyžaduje použitie tajného steganografického kľúča a tajnej informácie. Generovanie krycieho videosúboru predstavuje funkciu

$$X(A, D)$$

kde  $X$  je funkcia, ktorá generuje krycí videosúbor na základe tajnej informácie. Parameter  $D$  predstavuje bity vkladanej tajnej informácie a parameter  $A$  predstavuje počet vzoriek potrebných pre ukrytie bitov  $D$ . [30] [13]

Tieto metódy vyžadujú na vstup databázu obrázkov, z ktorých sa zhromaždí požadovaný počet pre vygenerovanie krycieho videa. Výhodou takýchto metód je, že útočníkovi neposkytujú pôvodné obrázky. Naopak, nevýhodou je, že technika môže vyvolať podozrenie u útočníka, ak zhromaždená sekvencia obrázkov je voči sebe irelevantná. Alternatívou k tomu môže byť zhromaždenie obrázkov, ktoré budú tvoriť prezentáciu sprevádzanú zvukom. [30]