Acta Informatica Pragensia, 2015, 4(3): 258–275
DOI: 10.18267/j.aip.73
Peer-reviewed paper

Modifikácia steganografického algoritmu využívajúceho LSB použitím množiny stegomédií

Modification of Steganographic Algorithm Using LSB and a Set of Stegomedia

Branislav Madoš[*], Mária Feková[*]

Abstrakt

Ambícia získať možnosť ukrývať digitálne reprezentované informácie kódované ako sekvencie bitov v digitálnych krycích médiách je naplnená prostredníctvom viacerých steganografických algoritmov, vrátane algoritmu využívajúceho najmenej dôležitý bit jednotlivých dátových zložiek digitálnych médií – Least Significant Bit (LSB). Rozvinutím týchto algoritmov môže byť použitie viacerých krycích médií vo forme ich množín, do ktorých je ukrývaná informácia rozdistribuovaná pomocou viacerých distribučných funkcií. Predkladaný článok popisuje návrh steganografického algoritmu založeného na využití najmenej dôležitého bitu digitálnych médií (LSB) a definovaní troch distribučných funkcií. Súčasťou článku je aj predstavenie programového vybavenia, ktoré bolo v tejto súvislosti navrhnuté, implementované a otestované.

Klíčová slova: Steganografia, Least Significant Bit, LSB, multi-carrier.

Abstract

Ambition to achieve possibility to hide digitally represented information which is coded in bit sequences into digital cover media is fulfilled through a number of steganographic algorithms, including Least Significant Bit (LSB) algorithm. A further development of those algorithms can be seen in the use of multiple cover media in the form of their sets, into which digital information is distributed by the use of multiple distribution functions (multi-carrier steganographic algorithms). This paper describes design of steganographic algorithm that is based on the use of the Least Significant Bit (LSB) and three distribution functions, which allow to distribute digital information into the set of cover media. The part of this article is describing software solution which was designed, developed and tested as the part of this research.

Keywords: Steganography, Least Significant Bit, LSB, Multi-carrier.

[*] Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic
✉ branislav.mados@tuke.sk, maria.fekova@student.tuke.sk

1 Úvod

Steganografia je súčasťou techník špecializovaných na ukrývanie informácií, pričom sú pri nej používané komunikačné kanály umožňujúce utajenú komunikáciu na pozadí otvorenej, neutajenej komunikácie. Bližšie sa problematike utajovaných komunikačných kanálov venuje Reiland (2005). Príbuznou problematikou je watermarking, ktorý sa líši od steganografie cieľom použitia vkladania informácie a snahou o robustnosť používaných algoritmov v zmysle nemožnosti odstránenia vloženej informácie manipuláciou s použitými médiami. Vzťahu watermarkingu a steganografie sa venuje Cox (2008) a Baran et al. (2001).

Steganografiu možno deliť podľa viacerých kritérií, vynikajúci prehľad tejto problematiky ponúka Petitcolas et al. (1997) a Kessler and Hosmer (2011). Klasické delenie steganografie predstavuje jej rozdelenie na predigitálnu éru a éru digitálnu, založenú na používaní výpočtových prostriedkov. Príkladom môže byť lingvistická steganografia, založená na využití metód ukrývajúcich informácie do klasického prirodzeného ľudského jazyka, často využívajúc krycie médiá v podobe písaného textu. Bližšie Chapman et al. (2001) a Bennet (2004). Steganografia dokáže využiť aj programovacie jazyky, ukrývanie informácií v podobe škodlivého kódu v programovom kóde pomocou obfuskácie rozoberá Hurtuk et al (2014) a Hurtuk (2014). Technická steganografia ukrýva správy pomocou technických zariadení do rôznych typov médií, príkladom môže byť neviditeľný atrament, mikrobodka apod.

Moderná steganografia patriaca do digitálnej éry využíva s výhodou možnosti výpočtovej techniky. Možno ju klasifikovať napríklad podľa druhov použitých médií. Príkladom môžu byť texty, obrázky, audiosekvencie alebo videosekvencie. Steganografia založená na použití obrázkov môže byť rozdelená na rastrovú a vektorovú. Medzi základné typy steganografie využívajúce rastrové obrazové formáty patrí steganografia s využitím najmenej dôležitého bitu – Least Significant Bit (LSB), pričom LSB môže byť aplikovaná aj na iné typ multimédií ako je napríklad zvuk, tak ako to ukazuje Cvejic a Seppanen (2014). Medzi steganografické postupy využívajúce vektorovú grafiku patria algoritmy využívajúce jittering a embedding. Jittering je podobný LSB steganografii, keď umožňuje ukrývať utajované správy do najmenej dôležitých číslic číselných hodnôt popisujúcich obraz. Embedding ukrýva utajovanú informáciu pridávaním ďalších, redundantných číselných hodnôt popisujúcich obraz tak, aby sa jeho vizuálna reprezentácia nezmenila. Algoritmus využívajúci embedding vo vektorovom formáte SVG prináša Madoš (2014).Steganografia s použitím najmenej dôležitého bitu predstavuje jednoduchý a ľahko programovo implementovateľný koncept, ktorý je dobre preskúmaný tak z hľadiska možností jeho využitia ako aj z hľadiska steganalytického, teda odhaľovania jeho použitia. Spoľahlivou detekciou LSB steganografie sa zaoberá Fridrich et al. (2001).

Predkladaný článok sa zaoberá v druhej kapitole predstavením LSB substitučného algoritmu v spojení s rastrovými obrazovými formátmi. V nasledujúcej kapitole predstavuje návrh modifikácie LSB algoritmu použitím viacerých krycích médií rôznych typov (multi-carrier) a doplnením troch distribučných funkcií, umožňujúcich výber spôsobu distribúcie jednotlivých bitov ukrývanej správy do jednotlivých krycích médií, výber poradia jednotlivých krycích médií a spôsob uloženia jednotlivých bitov ukrývanej správy v každom z krycích súborov. Štvrtá kapitola predstavuje hlavné črty programovej implementácie navrhnutej modifikácie LSB algoritmu. Nasledujúca kapitola predstavuje zhrnutie časti testov, ktoré boli po implementačnej fáze programového vybavenia realizované, pričom sa zaoberá testami, ktoré súviseli s použiteľnosťou aplikačného programového vybavenia z hľadiska dĺžky času vykonávania jeho kľúčovej funkcionality. Posledná kapitola článku, predstavujúca jeho záver, zhŕňa výhody a nevýhody navrhnutej modifikácie algoritmu.

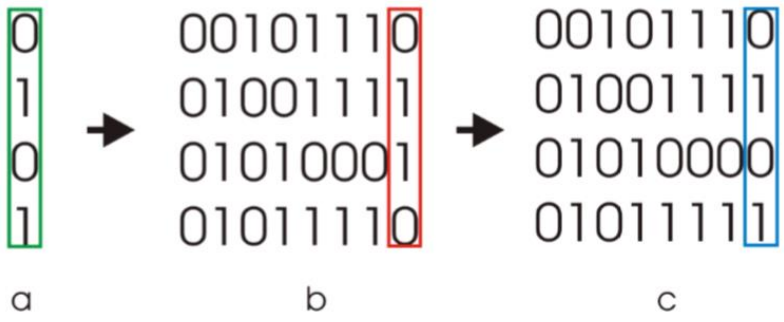
2 Klasický Least Significant Bit algoritmus

LSB substitučná steganografia (z anglického Least Significant Bit) nad rastrovými obrázkami využíva na vkladanie tajnej správy krycie médium v podobe obrázka, kde s každým pixelom potenciálneho nosiča správ je zviazaná informácia o jeho polohe v obrázku a farbe.

Farba pixela rastrového obrázka je najčastejšie kódovaná v prípade monochromatických obrázkov ako 8-bitový vektor, ktorého hodnota je kódom príslušného odtieňa farby zo škály sivej, obsahujúcej celkovo 256 rôznych odtieňov. V prípade farebných obrázkov je farba pixelu zložená z väčšieho počtu farebných zložiek, pričom každá farebná zložka je kódovaná ako samostatný bitový vektor. Farba pixelu sa v prípade plnofarebných obrázkov skladá najčastejšie z troch zložiek, označených ako R, G a B (z anglického Red, Green a Blue) pre červenú, zelenú a modrú farebnú zložku, pričom každá je osobitne kódovaná najčastejšie ako 8-bitový vektor. Farba jedného bodu je následne definovaná pomocou 24 bitov, čo znamená 16 777 216 odtieňov farieb. Rozsahy kódov odtieňov farieb tak pre monochromatické, ako aj pre plnofarebné obrázky umožňujú tak veľkú farebnú škálu, že ľudské oko nie je schopné rozoznať dva odtiene farby, ktorých bitové vektory sa líšia iba o hodnotu 1, to znamená v bite s najnižším významom.

Posledný, najmenej signifikantný bit (odtiaľ Least Significant Bit - LSB) je využitý pre zápis ukrývanej informácie, ktorá nahrádza posledný bit kódu príslušnej farby. Štatisticky je príslušný bit správy s 50% pravdepodobnosťou zhodný s pôvodným bitom vektora farby, preto je zamieňaných v priemere iba 50% bitov, tak ako to ukazuje Obr. 1, kde je v zelenom obdĺžniku vyznačený prúd 4 bitov, z ktorých každý je zapisovaný do najmenej dôležitého bitu príslušného pixela obrázka (tieto bity sú vyznačené červeným obdĺžnikom). Výsledkom je zmena týchto bitov, tak ako je vyznačené na Obr. 1 vpravo modrou farbou. Porovnaním bitov vyznačených červeným obdĺžnikom a modrým obdĺžnikom je možné vidieť, že 50% bitov bolo zmenených a 50% má pôvodnú hodnotu.V prípade plnofarebných obrázkov je možné zmeniť najmenej dôležitý bit v jednej (ľubovoľnej) alebo viacerých farebných zložkách.

Pre uloženie utajovanej správy je používaný najmenej dôležitý bit vektora popisujúceho farebný odtieň pixela resp. jeho farebnú zložku, pretože tak ako stúpa hodnota rádu príslušného bitu, tak hodnota tohto bitu významnejšie ovplyvňuje celkovú hodnotu vektora, a teda aj odtieň zobrazovanej farby pixela. Zmeny vo vyšších rádoch vektora popisujúceho pixel by teba boli pre ľudské oko viditeľnejšie a ľahšie detegovateľné.



Obr. 1. *Utajovaná správa (a) ukrývaná do krycieho média (b), vzniká stegomédium (c). Zdroj: Autor.*

V prípade monochromatického obrazu je kapacita obrázka pre uloženie správy približne 1/8 z veľkosti obrázka na úložnom médiu, pretože 1/8 bitov je možné zmeniť. Toto sa na prvý pohľad môže javiť ako pomerne malá kapacita, avšak ako príklad možno uviesť

nekomprimovaný monochromatický BMP obrázok s rozlíšením 1 024 × 768 pixelov, čo je 786 432 pixelov, z ktorých farebný odtieň každého pixelu je kódovaný na 8-bitoch. Ak je nahradený 1 z 8 bitov, je možné pre zápis tajnej správy použiť 786 432 b, čo tvorí 98 304 B. To predstavuje pri kódovaní pomocou klasickej ASCII tabuľky 98 304 znakov, čo pri 1 800 znakoch na stranu predstavuje až 54,61strán textu. V prípade farebného obrázka s 24-bitovou farebnou hĺbkou je táto kapacita v ideálnom prípade až trojnásobná, čo znamená 163,84 strán textu.

V záujme čo najvyššieho utajenia správy sa odporúča používanie obrázkových médií s pestrými farbami a zložitými motívmi, nie s veľkými homogénnymi farebnými plochami, ako je napríklad čierne, prípadne biele pozadie. Zmeny v takýchto homogénnych plochách môžu byť pozorovateľné a v prípade nízkeho počtu bitov vektora popisujúceho každý pixel môžu spôsobovať dokonca vznik osamotených bodov dostatočne výraznej farby, ktoré môže odhaliť ľudské oko. Okrem vizuálnej steganalýzy, ktorú popisuje Watters (2008), keď sa vykonáva vizuálna inšpekcia obrazu buď v takej forme, v akej je obraz zachytený alebo vo forme, keď bol predspracovaný rôznymi technikami, ako ukazuje Davidson a Paul (2004), uľahčujúcimi odhalenie utajenej správy, existujú aj štatistické metódy, ktoré sú zamerané na odhaľovanie odchýlok v štatistických parametroch obrázka ako celku s cieľom určiť, či je daný obrázok stegomédiom. Štatistickou steganalýzou sa zaoberá Dumitrescu a Wu (2005). V prípade, ak je odhalená prítomnosť utajovanej správy, je možné pokračovať v snahe túto správu extrahovať. V mnohých prípadoch nie je k dispozícii pri prezeraní stegomédia pôvodný obrázok, ktorý nenesie utajovanú správu, čo ešte viac sťažuje možnosť identifikácie zmien v krycom médiu na úrovni najmenej významných bitov.

Pre maximalizáciu možného utajenia vloženej správy je možnéju pred jej ukrytím ešte zašifrovať, navyše nie je nutné správu uložiť lineárne v prúde pixelov, tak ako sú uložené v obrázku, ale je možné určiť niektorý z pixelov obrázka ako počiatočný a uložiť správu do za ním nasledujúcich pixelov, prípadne použiť zložitejší algoritmus pre výber konkrétnych pixelov, pričom ostatné pixely nebudú niesť žiadnu informáciu, alebo budú použité pre doplnenie informácie modifikujúcej štatisticky spracovávateľné parametre obrázka, čo umožní zmenšiť šancu na odhalenie použitia steganografie pomocou štatistickej steganalýzy.

3 Modifikovaný LSB algoritmus

V rámci tejto práce navrhnuté použitie LSB substitučného algoritmu sa pokúša nájsť jeho modifikáciu resp. rozšírenie, ktoré umožní zvýšiť bezpečnosť použitia tohto algoritmu a sťažiť jeho steganalýzu v zmysle zabránenia odhalenia stegomédia ako prostriedku pre prenos ukrytej správy, ako aj zabránenia prípadnej extrakcie ukrytej správy v prípade, ak bolo použitie stegomédia odhalené.

3.1 Množina krycích médií

V práci navrhnutý modifikovaný LSB algoritmus nevyužíva pre ukrytie informácie iba jedno krycie médium, ale používa celú množinu K pozostávajúcu z n krycích médií (súborov) označených ako S_1, S_2, S_3 až S_n . Možno teda vyjadriť, že

$$K = \{ S_1, S_2, S_3, \ldots, S_n \}$$

Množina krycích médií nemusí nevyhnutne pozostávať výlučne zo súborov jediného typu, napríklad z rastrovej grafiky vo formáte bmp, alebo png, môžu byť využité akékoľvek multimediálne formáty, na ktoré môže byť aplikovaný LSB algoritmus, príkladom môžu byť zvukové vzorky vo formáte wav. Súčasťou množiny K tak môže byť napríklad jeden rastrový obrázok vo formáte png, jeden obrázok vo formáte gif a jedna zvuková vzorka vo formáte wav.

Krycie médiá nemusia mať rovnakú veľkosť v zmysle počtu bitov, z ktorých sa skladajú, resp. počtu bitov, ktoré je možné využiť pre uloženie ukrývanej správy. Pre celkovú kapacitu *C*množiny *K* krycích médií určenej pre ukrytie stegosprávy však neplatí, že sa dá vypočítať ako jednoduchý súčet kapacity jednotlivých krycích médií. Možno ju vypočítať ako

$$C = min(K) \times n \; [b]$$

kde

min(K) je najmenšia z kapacít jednotlivých stegomédií z množiny *K,*

n je počet stegomédií v množine *K*

Ak je kapacita stegomédií množiny *K*nedostatočná pre uloženie stegosprávy, je možné upraviť celkovú kapacitu množiny *K* viacerými spôsobmi, medzi ktoré patrí napríklad:

- Rozšírenie množiny *K* o ďalšie stegomédium, pričom ak je jeho kapacita zhodná alebo vyššia ako je *min(K)*, rozšíri sa kapacita celej množiny *K* práve o túto hodnotu *min(K)*. Ak však toto médium bude mať menšiu kapacitu ako je *min(K)* je potrebné zvážiť, či prispeje svojou kapacitou v dostatočnej miere, k celkovej kapacite súboru médií *K,* pretože vznikne nová, menšia hodnota *min(K)*.
- Zámenou najmenšieho média z množiny za iné médium s väčšou kapacitou sa súčasne zväčší aj celková kapacita množiny *K* na novú hodnotu $min_1(K) \times n$. Táto úprava súboru používaných médií môže pomôcť k zvýšeniu jeho celkovej kapacity najmä vtedy, ak má jedno z médií výrazne menšiu kapacitu ako všetky ostatné.
- Vynechaním najmenšieho média z množiny *K* sa môže zvýšiť celková kapacita množiny *K* na novú hodnotu $min_1(K) \times (n - 1)$, keď $min_1(K)$ je hodnota aktuálne najmenšieho stegomédia z množiny *K*. Počet médií v tejto množine v tomto prípade síce klesne o 1, preto je vo vzorci využitá hodnota $n - 1$, ale zvýšenie hodnoty *min(K)* môže túto zmenu vykompenzovať a celková kapacita *C* môže stúpnuť. Je však potrebné zvážiť, či touto úpravou nedôjde k navýšeniu celkovej kapacity množiny *K* o menšiu hodnotu ako je kapacita vynechaného média, teda či nedôjde k celkovému poklesu kapacity.

V prípade, ak sú pre ukrývanie dát používané súbory obsahujúce farebnú rastrovú grafiku s viacerými farebnými zložkami, je možné využívať pre uloženie dát jednu, viacero alebo všetky farebné zložky obrazu. V prípade klasického RGB modelu je teda možné použiť nie len jednu, ale dve prípadne tri farebné zložky. Ak je teda v množine *K* použitý takýto rastrový obrázok, je možné upraviť vzorec pre výpočet celkovej kapacity množiny médií dvoma spôsobmi:

- Kapacita súboru nesúceho rastrovú grafiku v pixeloch sa vynásobí počtom farebných zložiek modelu, ktoré budú použité pre uloženie stegosprávy, čo môže mať vplyv aj na hodnotu*min(K)*celej množiny *K* krycích médií a môže teda túto hodnotu zdvojnásobiť

alebo strojnásobiť, na druhej strane, ak nejde o médium, ktoré má v celom súbore minimálnu kapacitu, nemusí sa táto úprava nijako prejaviť na celkovej kapacite celého súboru stegomédií.

- Každá z farebných zložiek RGB modelu takejto rastrovej grafiky sa bude považovať za samostatný súbor, a teda zvýši sa počet stegomédií n v množine K na hodnotu (n + 1) pri použití dvoch farebných zložiek, resp. (n + 2) pri použití troch farebných zložiek.

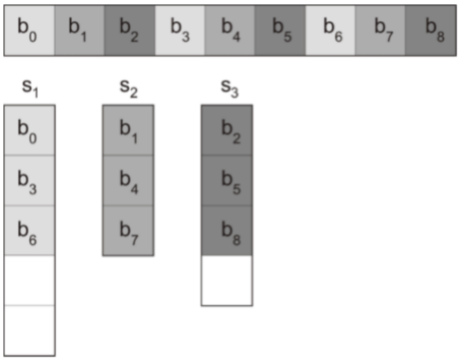
3.2 Distribučná funkcia f_1

Ukrývaná správa je pred jej uložením do množiny *K* krycích médií predspracovaná, pričom v tom čase je už známe koľko multimediálnych súborov bude túto množinu tvoriť a známa je aj minimálna kapacita stegomédií *min(K)*, a teda aj celková kapacita tejto množiny súborov *C*pre uloženie ukrývanej správy. Podstatou predspracovania stegosprávy je aplikovanie postupu, ktorý sekvenciu bitov ukrývanej správy rozdelí do súboru *L* množín s celkovým počtom *m*, ktorý je zložený z množín s_1, s_2, s_3, ... , s_m:

$$L = \{ s_1, s_2, s_3, \dots , s_m \}$$

Algoritmus na tieto účely zavádza distribučnú funkciu f_1, ktorá každému bitu ukrývanej správy priradzuje konkrétnu množinu S_x zo súboru množín *L*, do ktorej bude príslušný bit patriť. Výber konkrétnej realizácie distribučnej funkcie môže byť ponechaný na konkrétnu programovú implementáciu tohto algoritmu, prípadne môže byť implementovaných viacero takýchto verzií distribučnej funkcie a výber konkrétnej z nich môže byť súčasťou stegokľúča. Ponechanie tejto voľnosti na druhej strane znamená, že programové implementácie tohto algoritmu s rôznymi verziami distribučných funkcií nemusia byť navzájom kompatibilné a v prípade viacerých implementovaných distribučných funkcií v rámci jednej programovej implementácie musí byť výber konkrétnej z nich ponechaný na používateľa, prípadne mu musí byť daná k dispozícii informácia, ktorá z distribučných funkcií bola programom vybraná. Táto informácia musí byť potom súčasťou stegokľúča alebo musí byť súčasťou odosielanej správy. V prípade, ak je táto informácia súčasťou stegokľúča, musí byť známy tak odosielateľovi ako aj prijímateľovi stegosprávy. Zaslanie informácie o vybranej verzii distribučnej funkcie adresátovi správy však potom podlieha tým istým nevýhodám ako distribúcia šifrovacieho kľúča v prípade symetrického šifrovania. Ak je táto informácia súčasťou ukrývanej správy, musí byť zvolený spôsob jej kódovania a oddelenia od samotnej stegosprávy, aby bolo možné túto informáciu korektne extrahovať zo stegomédia a zabezpečiť jej správnu interpretáciu.

Jedna z najjednoduchších verzií distribučnej funkcie$f_1$rozdeľuje jednotlivé bity ukrývanej správy do množín tak, že prvý bit správy vloží do prvej množiny S_1, druhý bit vloží množiny S_2 až po *m-tý* bit, ktorý vloží do množiny S_m. Následne sa $m + 1$ bit vloží opäť do prvej, teda S_1 množiny atď., až kým nerozdistribuuje do množín všetky bity správy.



Obr.2. *Tajná správa distribuovaná do troch krycích médií distribučnou funkciou f_1.*
Zdroj: (Vokorokos et al., 2015)

Jednoduchý príklad takejto distribučnej funkcie je znázornený na Obr. 2, kde je správa, zložená zo sekvencie bitov b_0 až b_8, vložená do súboru množín L pozostávajúceho z troch množín S_1, S_2, a S_3, pričom bola zvolená v predošlom texte popísaná distribučná funkcia, ktorá cyklicky umiestňuje jednotlivé bity do jednotlivých množín tak, že bit b_0 uložila do množiny s_1, bit b_1 do množiny s_2, bit b_2 do množiny S_3 a pokračuje opäť množinou S_1, do ktorej ukladá bit b_3 atď. až kým, nie sú uložené všetky bity stegosprávy.

Kapacita pre uloženie stegosprávy je 5b pre médium S_1, 3b pre médium S_2 a 4b pre médium S_3. Celková kapacita množiny K médií sa dá vypočítať ako ich počet n = 3 v súčine s kapacitou toho média, ktoré ju má najnižšiu, a teda určuje hodnotu *min(K)*. V toto prípade je to médium S_2, ktoré má kapacitu 3b. Celková kapacita súboru médií je potom 3 x 3b, teda 9b.

3.3 Distribučná funkcia f_2

Druhá distribučná funkcia sa týka poradia stegomédií, teda ako sú priradené množiny S_1, S_2, S_3, až S_m jednotlivým súborom, resp. jednotlivým farebným zložkám v súboroch, ak sú považované za samostatné súbory. Funkcia f_2 teda každej množine $S_x \in L$ priradzuje konkrétny krycí súbor resp. jeho konkrétnu farebnú zložku.

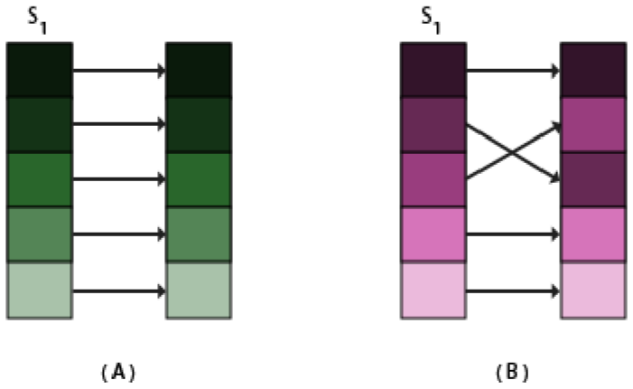
Opäť, ako v prípade prvej distribučnej funkcie môže byť aj tu výber poradia krycích médií resp. farebných zložiek ponechaný na používateľovi aplikácie, a teda sa tento výber stáva súčasťou stegokľúča, ktorý musí poznať tak odosielateľ, ako aj prijímateľ správy. Alternatívne je možné výber poradia ponechať na konkrétnej programovej implementácii algoritmu. Pre voľbu poradia by potom bolo možné využiť viaceré parametre krycích médií ako je napríklad ich veľkosť, prípadne názov a bolo by ich možné napríklad zoradiť od najväčšieho po najmenší, prípadne vzostupne či zostupne podľa abecedy na základe ich názvu, pričom je možné použiť na tieto účely rôzne metriky reťazcov. Poradie obrázkov možno zostaviť aj na základe ich obsahu, teda motívu, ktorý zobrazujú. To v súčasnosti nie je možné účinne automatizovať a je potrebná asistencia používateľa, ktorý sám určí obsah obrázkov a následne zostaví ich poradie.

3.4 Distribučná funkcia f_3

Tretia distribučná funkcia je použitá pre distribúciu jednotlivých bitov z príslušnej množiny S_x v rámci daného súboru, resp. v rámci konkrétnej farebnej zložky. Využité môžu byť všetky bity krycieho média až do kapacity *min(K)* alebo môže verzia distribučnej funkcie rozhodnúť, že niektoré bity nebudú zmenené a zmenší sa aj hodnota *min(K)*a celková kapacita súboru krycích médií C. Distribučná funkcia potom môže jednotlivé bity zapisovať v poradí, v akom sa nachádzajú v príslušnej množine S_x alebo môže predstavovať algoritmus, pri ktorom môže

byť toto poradie pozmenené, čo predstavuje ďalšiu komplikáciu pre prípadného útočníka, ktorý by chcel diagnostikovať prítomnosť ukrytej stegosprávy na základe analýzy jej obsahu, ako aj na základe analýzy globálnych štatistických parametrov konkrétneho stegomédia.

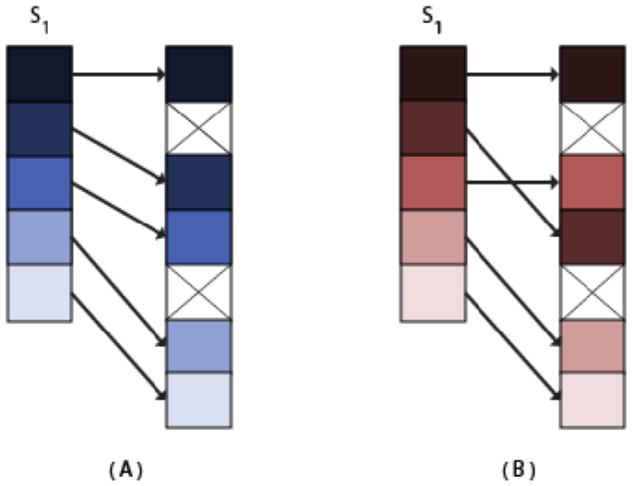
Na Obr. 3. je znázornená situácia, keď distribúcia jednotlivých bitov správy v rámci súboru, resp. farebnej zložky prebieha tak, že jednotlivé bity správy sú ukladané bezprostredne po sebe. Takto je možné využiť celú kapacitu stegomédia, resp. jeho kapacitu až do veľkosti *min(K)*. Na Obr. 3(A) je zobrazená verzia distribučnej funkcie, ktorá je najjednoduchšia a ukladá jednotlivé bity sekvenčne, to znamená, že ich poradie v sekvencii v množine S_x a v stegomédiu je totožné. Verzia distribučnej funkcie na Obr. 3(B) ukladá predpripravenú sekvenciu bitov tak, že poradie jednotlivých bitov sa po ich uložení do stegomédia líši.



Obr. 3. *Distribučná funkcia f_3 využívajúca plnú kapacitu krycích médiímin(K).*
Zdroj: (Vokorokos et al., 2015).

Na Obr. 4. je zobrazená situácia, keď zvolená verzia distribučnej funkcief_3 nevyužíva všetky bity stegomédia z kapacity *min(K)*. Nevýhodou takejto verzie distribučnej funkcie je, že nedokáže využiť kapacitu stegomédia až do hodnoty *min(K)* a znižuje tak aj celkovú kapacitu súboru médií *C*, avšak na druhej strane pôsobí pozitívne, pretože zvyšuje pravdepodobnosť, že nedôjde k detegovaniu použitia steganografického algoritmu a nedôjde ani k extrahovaniu správy. Jednotlivé bity, ktoré nie sú využité pre zápis ukrývanej informácie potom môžu zostať v pôvodnej verzii alebo môžu byť modifikované tak, aby vhodne upravovali štatistické hodnoty týkajúce sa obrázku ako celku.

Na Obr. 4(A) je zobrazená verzia distribučnej funkcie, ktorá ukladá jednotlivé bity sekvenčne, to znamená, že ich poradie v predpripravenej sekvencii S_x a v stegomédiu je zhodné. Verzia distribučnej funkcie na Obr. 4(B) ukladá predpripravenú sekvenciu bitov tak, že poradie jednotlivých bitov sa po ich uložení do stegomédia líši.



Obr. 4. *Distribučná funkcia f_3 využívajúca iba niektoré bity stegomédia. Zdroj: (Vokorokos et al., 2015)*

Zrealizovať by bolo možné aj algoritmus, ktorý využíva rozdielne verzie distribučnej funkcie f_3 v rámci jedného súboru stegomédií, resp. v rámci súboru farebných zložiek. To by na jednej strane sťažilo získanie informácie, či bola použitá steganografia a ešte viac by skomplikovalo extrahovanie ukrytej správy, na druhej strane by si to vyžadovalo použitie ďalšej doplnkovej informácie, ktorá by musela byť súčasťou kľúča, a teda dopredu dohodnutá alebo by musela byť distribuovaná spolu s ukrývanou správou, pretože ku každému súboru by osobitne musela existovať informácia, ktorá konkrétna distribučná funkcia f_3 bola použitá.

Ak by bola aplikovaná na každé stegomédium zo súboru iná distribučná funkcia f_3, bolo by súčasne možné využiť zostávajúcu kapacitu tých súborov, ktorých kapacita je väčšia ako *min(K)* a to tak, že táto voľná kapacita by bola rozdistribuovaná medzi bity ukrývanej správy a využitá by bola na úpravu globálnych štatistických parametrov stegomédia uložením vhodných bitov, nenesúcich žiadnu informáciu. Čím väčší by potom bol rozdiel medzi celkovou kapacitou konkrétneho média a hodnotou *min(K)*, tým väčší podiel bitov stegomédia by bolo možné použiť na tento účel, a tým viac by bolo možné zo štatistického hľadiska zakryť existenciu stegosprávy a použitie steganografického algoritmu.

Jednotlivé stegomédiá sú určené na distribúciu prostredníctvom rôznych komunikačných kanálov, pričom prijímateľ správy musí tieto komunikačné kanály poznať. Použitím viacerých komunikačných kanálov sa znižuje pravdepodobnosť možnosti extrakcie správy v prípade odhalenia jedného z nich, pretože prípadný útočník by musel odhaliť všetky komunikačné kanály.

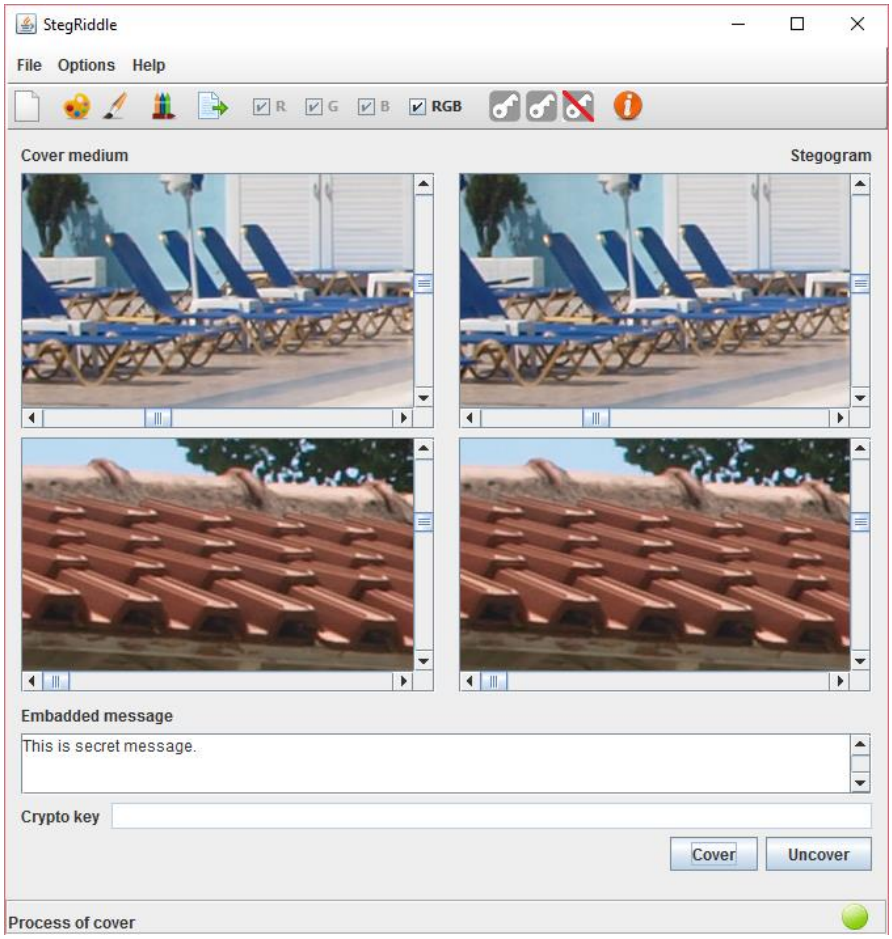
4 Programová implementácia navrhnutého algoritmu

Programové vybavenie implementujúce navrhnutú modifikáciu LSB algoritmu bolo realizované v programovacom jazyku JAVA s využitím plne grafického používateľského rozhrania. V strednej časti hlavného okna aplikácie bol vyčlenený priestor pre zobrazenie dvoch pôvodných rastrových obrázkov a ďalších dvoch obrázkov, ktoré vzniknú ich modifikáciou uložením stegosprávy. Z hľadiska použiteľnosti programu boli zvažované viaceré možnosti ako zobrazovať obrázky v obmedzenom priestore, ktorý bol na tento účel vyčlenený. Jedna z možností bola, že sa obrázok svojimi rozmermi prispôsobí vyčlenenej ploche, to však znamená, že môže dochádzať k deformácii obrázka, keď pomery jeho strán a priestoru určeného na jeho zobrazenie nie sú zhodné. Navyše, ak má obrázok veľké rozlíšenie, pri jeho zobrazení na malej ploche dochádza k jeho výraznému zmenšeniu.

Vzhľadom k tomu, že zobrazenie obrázkov má umožniť používateľovi vizuálnu inšpekciu s cieľom rozoznať aj minimálne, voľným okom rozoznateľné zmeny vo farebných odtieňoch jednotlivých pixelov, bolo by toto potenciálne výrazné zmenšenie obrázkov nevhodné. Alternatívne je možné zobraziť obrázky, ktorých rozlíšenie je väčšie ako plocha, ktorá je k dispozícii na zobrazovanie týchto obrázkov tak, že bude zobrazený iba výrez z obrázka v jeho aktuálnej veľkosti a používateľ dostane k dispozícii vertikálny a horizontálny posuvník pre každý zobrazený obrázok, aby mohol interaktívne vyberať zobrazený výrez obrázka a posúvať tento výrez tak v horizontálnom, ako aj vertikálnom smere. Ako sa ukázalo počas práce s prototypom aplikácie, tento spôsob zobrazovania obrázkov je vyhovujúci, preto bol implementovaný v jej konečnej verzii. Po načítaní sa obrázky zobrazia v ľavej časti určenej plochy (označená ako Covermedium na Obr. 5). Ak je zvolená steganografická funkcionalita aplikácie, tak sú následne stegogramy zobrazené v pravej časti určenej plochy (označená ako Stegogram na Obr. 5). Ak je však využívaná steganalytická funkcia, tak ostane pravá časť plochy prázdna.

Ďalšou skupinou ovládacích prvkov aplikácie, ktorá bola uložená v dolnej časti jej hlavného okna, je textové pole určené pre interaktívne zadávanie ukrývanej správy alebo pre zobrazenie extrahovanej správy a textové pole pre vkladanie kryptokľúča, ktorý je používaný ako kľúč pre použité šifrovanie.

Navrhnutá aplikácia má v súlade s vyššie uvedeným návrhom implementované príslušné verzie všetkých troch distribučných funkcií.



Obr. 5. *Grafické používateľské rozhranie aplikačného programového vybavenia implementujúceho navrhnutú modifikáciu LSB algoritmu s využitím dvoch krycích médií. Zdroj: Autor.*

Prvá distribučná funkcia rozhoduje o tom ako budú jednotlivé bity ukrývanej správy rozdistribuované do jednotlivých stegomédií. Navrhnutá aplikácia obmedzuje dátový formát stegomédií na rastrovú plnofarebnú grafiku vo formáte png a bmp. Aplikácia počas načítavania potenciálneho stegomédia kontroluje, či stegomédium je rastrovým obrázkom, či je plnofarebné a či disponuje troma farebnými zložkami RGB s počtom bitov 8 na jednu farebnú zložku pixelu. Používateľ môže vybrať, či bude používať jednu, dve alebo všetky farebné zložky. Vzhľadom k tomu, že sú používané dve stegomédiá, počet množín, do ktorých je rozdelená stegospráva je rovný 2 × počet použitých farebných zložiek. Nie je teda možné vybrať rozdielny počet farebných zložiek v jednom a druhom stegomédiu. Distribučná funkcia rozdeľuje jednotlivé bity tak, že ak sú napríklad vybrané zložky R a B, tak celkový počet použitých farebných zložiek n je rovný štyrom. Prvý bit správy je potom vložený do farebnej zložky R prvého obrázku, druhý bit je vložený do zložky R druhého obrázku, tretí bit je vložený do B zložky prvého obrázku, štvrtý bit je vložený do B zložky druhého obrázku. Piaty bit je potom znova vkladaný do farebnej zložky R prvého krycieho média a takýmto spôsobom sú následne rozdistribuované aj všetky ostatné bity ukrývanej správy.

Druhá distribučná funkcia, určujúca poradie krycích médií, resp. ich farebných zložiek, zoraďuje krycie súbory podľa poradia ich načítania. Toto poradie určuje používateľ pri ukrývaní správy ako aj pri jej extrakcii, je teda súčasťou stegokľúča.

Tretia distribučná funkcia je implementovaná v najjednoduchšej forme, teda bity správy sú vkladané do krycích médií v rovnakom poradí ako sú v množinách, ktoré vytvorila prvá distribučná funkcia, a do bezprostredne za sebou nasledujúcich pixelov.

Šifrovanie nie je priamou súčasťou steganografických postupov avšak veľmi často sa obidva prístupy k utajenej komunikácii používajú súčasne, keď je steganografická správa, ktorej existencia má byť utajená vložením do stegomédia pred týmto ukrytím ešte zašifrovaná. Požívateľovi steganografického softvéru nebráni nič v tom, aby na zašifrovanie správy použil špecializovaný programový nástroj určený primárne na účely šifrovania. Pre zvýšenie používateľského komfortu steganografických aplikácií sa však bežne do tohto softvéru implementujú aj kryptografické funkcie. To umožňuje inštalovaním jedinej aplikácie pokryť obidve oblasti ukrývania dát. Preto bola takáto funkcionalita zahrnutá aj do navrhnutej aplikácie. Zašifrovanie správy navyše môže napomôcť zvýšiť efektivitu použitého steganografického algoritmu, a to v prípade, ak je steganalýza založená na vyhľadávaní zmysluplných fragmentov ukrytej správy.

Implementované boli dva algoritmy šifrovania AES a 3DES, pričom prvý využíva 128 bitový kľúč v podobe 16 ASCII znakov v 8 bitovom kódovaní, druhý z vybraných šifrovacích algoritmov využíva 168 bitový kľúč, ktorý je reprezentovaný 21 ASCII znakmi v 8 bitovom kódovaní.

5 Testovanie navrhnutého programového vybavenia

Realizácia hlavnej funkcionality aplikácie, teda ukrývanie tajnej správy do krycieho média, nebola počas vývoja aplikácie vyhodnotená ako zdĺhavá, preto neboli používané optimalizačné metódy pre zníženie jej časovej náročnosti. Potrebný čas na vykonávanie funkcií aplikácie sa napriek tomu líši od použitia k použitiu a závisí ako od veľkosti použitých krycích médií, tak aj od veľkosti vkladanej tajnej správy.

Cieľom testovania bolo zistiť aká je závislosť času potrebného pre ukrytie správy a prípadného šifrovania správy od veľkosti použitého média pri konštantnej veľkosti správy. Testovaný bol aj opačný proces, to znamená extrakcia správy a jej dešifrovanie, pričom

sledovaný bol takisto čas potrebný na realizáciu týchto operácií. Na testovanie bol použitý stolný počítač s procesorom Intel Core i5 4460 3.4GHz, RAM 8GB DDR3, Windows 8 64-bit.

Pre otestovanie bolo náhodne vybraných 10 krycích médií rôznych veľkostí, do ktorých bola vkladaná tajná správa, ktorá bola pre jasnejšiu viditeľnosť závislosti vo forme súboru konštantnej veľkosti. Systematicky bola vkladaná tajná správa prv do jednej zložky RGB modelu, potom do dvoch a nakoniec do troch, pričom bolo taktiež rozlišované, či nebola použitá kryptografia alebo bol použitý šifrovací algoritmus AES, resp. šifrovací algoritmus 3DES.

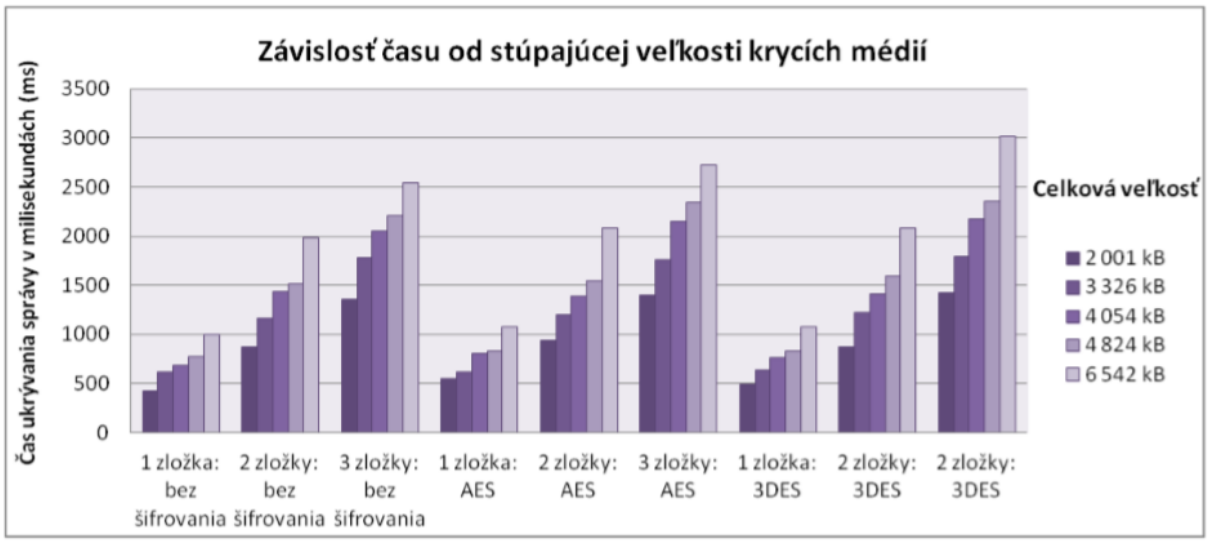
Všetky rôzne varianty boli sledované na 5-tich dvojiciach náhodne vybraných krycích médií. Keďže časová závislosť od veľkosti krycích médií bola pozorovateľná už pri prvých desiatich náhodných krycích médiách, je možné vyvodiť záver, že so vzrastajúcou veľkosťou krycích médií bude vzrastať aj čas potrebný na ukrytie správy do takýchto médií.

Tab. 1. *Graf závislosti času spracovávania v milisekundách od veľkosti krycích médií. Zdroj: Autor.*

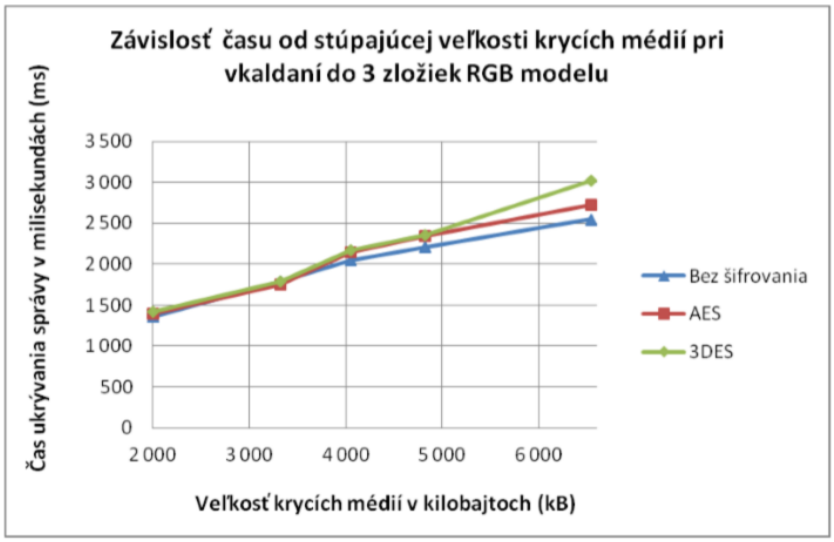
Veľkosť správy	Názov	Veľkosť obrázka	Rozmery	1 zložka RGB	2 zložky RGB	3 zložky RGB	Čas procesu - COVER			Čas procesu - UNCOVER		
							Bez šifrovani	AES	3DES	Bez šifrovani	AES	3DES
55,0 kB	1.bmp	1 134 kB	746 x 518	x			428 ms	543 ms	489 ms	295 ms	1 142 ms	1 212 ms
	2.bmp	867 kB	800 x 600		x		877 ms	936 ms	872 ms	592 ms	1 361 ms	1 618 ms
		2 001 kB				x	1 357 ms	1 398 ms	1 424 ms	1 107 ms	1 553 ms	1 756 ms
55,0 kB	3.bmp	2 304 kB	1 024 x 768	x			614 ms	614 ms	635 ms	380 ms	1 248 ms	1 195 ms
	4.bmp	1 022 kB	671 x 519		x		1 161 ms	1 202 ms	1 216 ms	701 ms	1 499 ms	1 513 ms
		3 326 kB				x	1 785 ms	1 754 ms	1 790 ms	1 198 ms	1 747 ms	1 830 ms
55,0 kB	5.bmp	2 647 kB	2 000 x 754	x			685 ms	805 ms	764 ms	623 ms	1 548 ms	1 443 ms
	6.bmp	1 407 kB	800 x 600		x		1433 ms	1389 ms	1408 ms	1 185 ms	2 034 ms	1 962 ms
		4 054 kB				x	2047 ms	2148 ms	2170 ms	1 875 ms	2 505 ms	2 638 ms
55,0 kB	7.bmp	983 kB	670 x 500	x			774 ms	824 ms	827 ms	1 330 ms	1 340 ms	1 396 ms
	8.bmp	3 841 kB	1 280 x 1024		x		1 516 ms	1 545 ms	1 586 ms	1 938 ms	1 870 ms	1 769 ms
		4 824 kB				x	2 207 ms	2 345 ms	2 357 ms	2 346 ms	2 355 ms	2 463 ms
55,0 kB	9.bmp	2 701 kB	1 280 x 720	x			998 ms	1 071 ms	1 079 ms	1 465 ms	1 555 ms	1 549 ms
	10.bmp	3 841 kB	1 280 x 1024		x		1 989 ms	2 083 ms	2 085 ms	2 008 ms	2 126 ms	2 159 ms
		6 542 kB				x	2 547 ms	2 727 ms	3 019 ms	2 963 ms	2 864 ms	2 815 ms

Výsledky tohto testu znázorňuje Tab. 1., kde sú zhrnuté výsledky ukrývania informácie vždy do dvojice súborov a následne extrahovania informácie z týchto súborov. V tabuľke je indikované, či boli v konkrétnom teste použité pre uloženie informácie najmenej dôležité bity jednej, dvoch, alebo troch zložiek RGB modelu a či bolo použité šifrovanie a ak áno, aký algoritmus šifrovania bol využitý. Uvedené sú potom časy realizácie príslušnej operácie v milisekundách.

Z vykonaných testov vyplýva, že čím je celková veľkosť použitých médií väčšia, tým sa zväčšuje čas vykonávania procesu ukrývania resp. extrakcie správy. Veľkosť času potrebného na realizáciu príslušných operácií je však z hľadiska používateľského komfortu zanedbateľná. Názornejšie výsledky meraní zobrazuje graf vybranej časti výsledkov na Obr. 6. resp. Obr. 7. zobrazujúci časovú závislosť pri ukrývaní správ do krycích médií.



Obr. 6. *Graf závislosti času spracovávania v sekundách od veľkosti krycích médií. Zdroj: Autor.*



Obr. 7. *Graf závislosti času spracovávania od veľkosti krycích médií pri vkladaní tajnej správy do 3 RGB zložiek rastrového obrázka. Zdroj: (Vokorokos et al., 2015)*

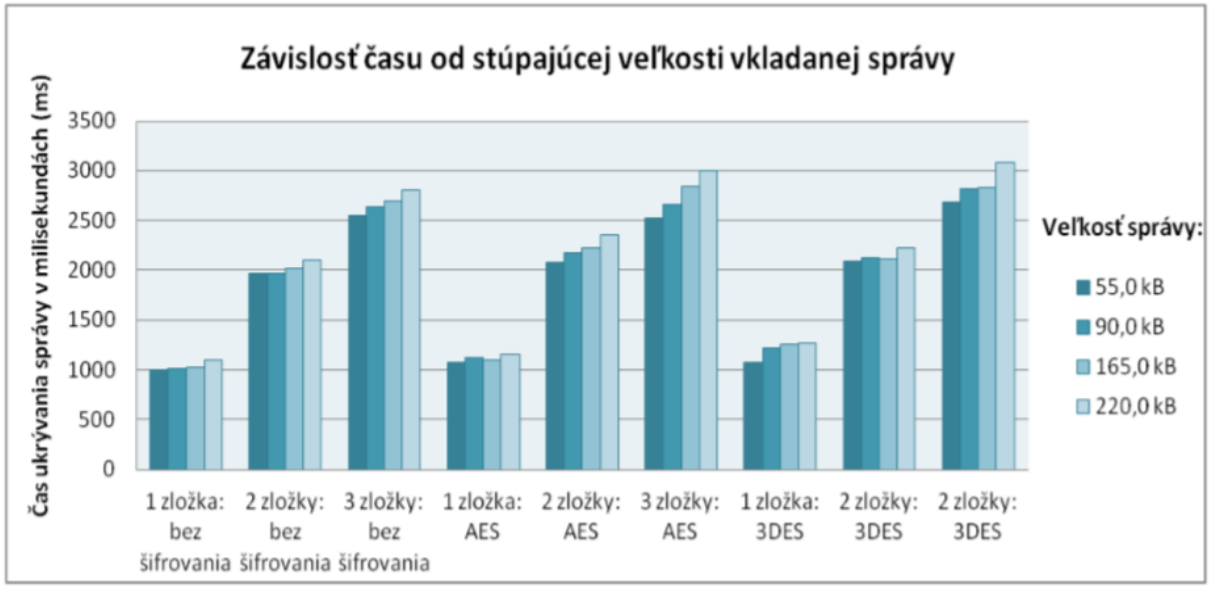
Ďalším krokom testovania závislosti trvania vykonávania funkcií aplikácie od veľkosti súborov bolo otestovanie závislosti dĺžky procesu ukrývania tajnej správy od veľkosti vkladanej správy. Takisto ako v prípade testovania závislosti času od stúpajúcej veľkosti krycích médií, tak aj v tomto prípade boli testované všetky kombinácie nastavení aplikácie, teda vloženie správy do jedného, dvoch alebo troch farebných kanálov, deaktivovanie použitia šifrovania, ako aj použitie šifrovacích algoritmov AES a 3DES. Pre názornejší pohľad na závislosť času od veľkosti správy bola vybraná dvojica krycích médií náhodnej veľkosti a postupne do nich boli vkladané tajné správy rôznych veľkostí. Výsledkom tohto testovania bolo zistenie, že narastanie veľkosti tajnej správy zvyšuje nároky na čas realizácie týchto funkcií aplikácie ako znázorňuje Tab. 2.

Tab. 2. *Graf závislosti času spracovávania v milisekundách od veľkosti ukrývanej informácie. Zdroj: Autor.*

Veľkosť správy	Názov	Veľkosť obrázka	Rozmery	1 zložka RGB	2 zložky RGB	3 zložky RGB	Čas procesu - COVER			Čas procesu - UNCOVER		
							Bez šifrovania	AES	3DES	Bez šifrovania	AES	3DES
55,0 kB	9.bmp	2 701 kB	746 x 518	x			998 ms	1 071 ms	1 079 ms	1 465 ms	1 555 ms	1 549 ms
	10.bmp	3 841 kB	800 x 600		x		1 969 ms	2 083 ms	2 085 ms	2 008 ms	2 126 ms	2 159 ms
		6 542 kB				x	2 547 ms	2 527 ms	2 689 ms	2 963 ms	2 864 ms	2 815 ms
90,0 kB	9.bmp	2 701 kB	746 x 518	x			1 008 ms	1 121 ms	1 214 ms	1 741 ms	1 345 ms	1 622 ms
	10.bmp	3 841 kB	800 x 600		x		1 971 ms	2 174 ms	2 124 ms	2 134 ms	2 046 ms	2 312 ms
		6 542 kB				x	2 632 ms	2 657 ms	2 817 ms	2 814 ms	2 768 ms	2 754 ms
165,0 kB	9.bmp	2 701 kB	746 x 518	x			1 021 ms	1 098 ms	1 254 ms	1 465 ms	1 479 ms	1 719 ms
	10.bmp	3 841 kB	800 x 600		x		2 014 ms	2 228 ms	2 118 ms	2 218 ms	2 224 ms	2 445 ms
		6 542 kB				x	2 699 ms	2 841 ms	2 825 ms	2 998 ms	2 955 ms	2 799 ms
220,0 kB	9.bmp	2 701 kB	746 x 518	x			1 102 ms	1 158 ms	1 261 ms	1 465 ms	1 614 ms	1 814 ms
	10.bmp	3 841 kB	800 x 600		x		2 100 ms	2 354 ms	2 227 ms	2 348 ms	2 255 ms	2 469 ms
		6 542 kB				x	2 810 ms	2 994 ms	3 079 ms	3 001 ms	2 924 ms	3 045 ms

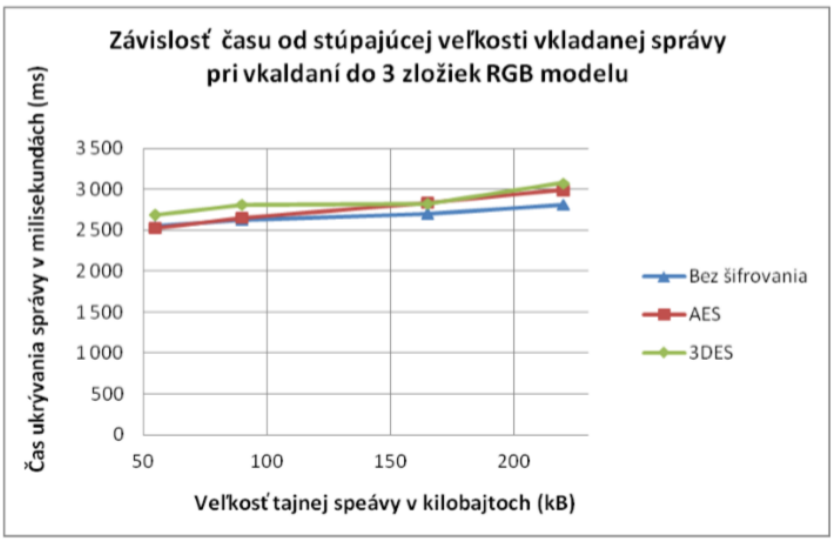
Výsledky meraní zaznamenané v tabuľke zobrazuje graf vybranej časti výsledkov na Obr. 8. zobrazujúci časovú závislosť pri ukrývaní správ do krycích médií.

.



Obr. 8. *Graf závislosti času spracovávania v milisekundách od veľkosti vkladanej utajovanej správy. Zdroj: Autor.*

Z grafu na Obr. 9. je možné získať informáciu, že v prípade ukrývania tajnej správy stúpajúcej veľkosti do krycích médií konštantnej veľkosti program vykazuje lineárnu závislosť času ukrývania a veľkosti tajnej správy. Súčasne možno skonštatovať, že tento čas je v prípade predpokladaných bežných spôsobov použitia programu akceptovateľný a nepôsobí negatívne z hľadiska použiteľnosti aplikácie.



Obr. 9. *Graf závislosti času spracovávania od veľkosti vkladanej správy do všetkých trochfarebných zložiek RGB modelu. Zdroj: (Vokorokos et al., 2015)*

6 Závery

Navrhnutá modifikácia LSB substitučného algoritmu prináša viacero výhod oproti použitiu klasického LSB algoritmu. Pri použití distribučných funkcií pri rozdeľovaní jednotlivých bitov ukrývanej správy je výhodou to, že ani jedno z použitých krycích médií nenesie úplnú utajovanú správu a nenesie ani sekvenciu bitov ukrývanej správy, ktoré by v nej nasledovali bezprostredne za sebou. Ak teda dôjde k odhaleniu ukrytia správy v jednom z n krycích médií, toto odhalenie neumožní extrakciu celej správy a ani žiadnej jej časti tak, aby ju bolo možné zmysluplne interpretovať. Ďalšou výhodou navrhnutého algoritmu je zväčšenie kapacity pre ukrytie utajovanej správy, keď oproti použitiu jedného stegomédia môže byť kapacita množiny n stegomédií s rovnakou veľkosťou až n-krát väčšia. Dosiaľ používané steganografické a steganalytické programy nie sú s týmto algoritmom kompatibilné, a teda neumožnia odhalenie ukrytia tajnej správy alebo prinajmenšom neumožnia extrahovanie ukrytej správy v prípade, ak odhalia samotné ukrytie správy.Použitie viacerých stegomédií umožňuje využitie viacerých komunikačných kanálov pre ich doručenie adresátovi, čo znižuje šancu útočníka odhaliť všetky tieto komunikačné kanály a získať tak všetky stegomédiá z použitej množiny. To mu značne sťaží až znemožní extrahovanie utajovanej správy v prípade odhalenia podmnožiny z použitej množiny stegomédií.

Nevýhodou navrhnutého algoritmu je, že celková kapacita množiny použitých krycích médií je pri ich celkovom počte n rovná n × kapacita najmenšieho stegomédia z množiny použitých stegomédií. Použitie jedného významne menšieho stegomédia tak môže znamenať výrazné zmenšenie kapacity celej množiny stegomédií. Použitie viacerých stegomédií, ako aj viacerých komunikačných kanálov, ktorými budú tieto stegomédiá zasielané adresátovi znamená zvýšenie rizika, že bude jedno alebo viacero použitých stegomédií narušených prípadne dôjde k narušeniu jedného alebo viacerých komunikačných kanálov, čo môže spôsobiť stratu stegomédia zasielaného týmto kanálom. Nemožnosť extrahovania utajovanej správy z neúplnej množiny stegomédií sa tak súčasne stáva aj nevýhodou tohto algoritmu.

Ak sa v ukrývanej správe bude periodicky vyskytovať rovnaký symbol, pri nevhodne zvolenom počte stegomédií sa môže tento symbol začať v jednom zo stegomédií opakovať s príliš vysokou frekvenciou, čo môže ohroziť utajenie časti ukrývanej správy uloženej v tomto stegomédiu. Následne môže prezradiť použitie daného komunikačného kanála,

a prípadne aj ostatných použitých stegomédií, čo môže napokon viesť až k extrakcii správy. Napríklad, ak by obsahom utajovanej správy mala byť séria binárne kódovaných desiatkových číslic (BCD kód) a pre ukrytie stegosprávy by boli použité štyri stegomédiá, negatívne by sa prejavila vlastnosť BCD kódu, ktorý v najvýznamnejšom bite kódu každej číslice obsahuje s 80% pravdepodobnosťou bit 0, to znamená, že pri štatisticky významnom počte číslic by sa táto pravdepodobnosť prejavila aj v súbore do ktorého by bol tento najvyšší bit ukladaný a znak 0 by sa v ňom vyskytoval v 80% prípadov.

Poďakovanie

Táto práca bola podporovaná Agentúrou na podporu výskumu a vývoja na základe zmluvy č. APVV-0008-10. Bola realizovaná na Katedre počítačov a informatiky, Fakulty elektrotechniky a informatiky, Technickej univerzity v Košiciach.

Zoznam použitých zdrojov

Baran, B., Gomez, S., & Bogarin, V. (2001). Steganographic Watermarking for Documents. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences* (pp. 1-10). Los Alamitos: IEEE.

Bennet, K. (2004). *Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text*. West Lafayette: Purdue University. CERIAS Tech Report 2004-13.

Cox, I. J. (2008). *Digital Watermarking and Steganography*. Second Edition. Burlington: Elsevier.

Cvejic, N., & Seppanen, T. (2005). Increasing Robustness of LSB Audio Steganography by Reduced Distortion LSB Coding. *Journal of Universal Computer Science*, 11(1), 56-65. doi: 10.3217/jucs-011-01-0056

Davidson, I., & Paul, G. (2004). Locating secret messages in images. In *ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 545-550). New York: ACM. doi: 10.1145/1014052.1014117

Dumitrescu, S., & Wu, X. (2008). LSB steganalysis based on high-order statistics. In *Proceedings of the 7th workshop on Multimedia and security* (pp. 25-32). New York: ACM. doi: 10.1145/1073170.1073176

Fridrich, J., Goljan, M., & Du., R. (2001). Reliable detection of LSB steganography in grayscale and color images. In *Proceedings of the 2001 workshop on Multimedia and security: new challenges* (pp. 27-30). New York: ACM. doi: 10.1145/1232454.1232466

Hurtuk, J., Čopjak, M., Dufala, M., & Drienik, P. (2014). The malicious code hiding techniques, code obfuscation problem. In *Proceedings of the 12th IEEE International Conference on Emerging eLearning Technologies and Applications* (pp. 181-185). Danvers: IEEE. doi: 10.1109/ICETA.2014.7107581

Hurtuk, J. (2014). Malware categorization and recognition problem. In: *Proceedings of the 14th Scientific Conference of Young Researchers* (pp. 207-211). Košice: TU.

Chapman, M., Davida, G., & Rennhard, M. (2001). A Practical and Effective Approach to Large-Scale Automated Linguistic Steganography. In *Proceedings of 4th International Conference ISC 2001* (pp. 156-165). New York: Springer. doi: 10.1007/3-540-45439-X_11

Kessler, G., & Hosmer, C. (2011). An Overview of Steganography. *Advances in Computers*, 83(1) 51-107.

Madoš, B., Hurtuk, J., Čopjak, M., Hamaš, P., & Ennert, M. (2014). Steganographic algorithm for information hiding using scalable vector graphics images. *Acta Electrotechnica et Informatica*, 14(4), 42-45. doi: 10.15546/aeei-2014-0040

Petitcolas, F. A. P., Anderson, R. J., & Kuhn, M. G. (1999). Information hiding a survey. *Proceedings of the IEEE, special issue on protection of multimedia content*, 87(7), 1062-1078.

Reiland, K., Oblitey, W., Ezekiel, S., Wolfe, J. (2005). *Steganography and Covert Channels*. Indiana: Indiana University of Pennsylvania.

Vokorokos, L., Madoš, B., Hurtuk, J., & Feková, M. (2015). Multi-carrier steganographic algorithm using LSB steganography. *Acta Electrotechnica et Informatica*, 15(2), 39-42. doi: 10.15546/aeei-2015-0016

Watters, P., Martin, F., & Stripf, H. S. (2008). Visual detection of LSB-encoded natural image steganography. *ACM Transactions on Applied Perception*, 5(1). doi: 10.1145/1279640.1328775