Napadi na steganografske sustave

Veselovac, Gabriel

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

Napadi na steganografske sustave

Završni rad

Gabriel Veselovac

Osijek, 2022.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 31.08.2022.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Gabriel Veselovac
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	R 4437, 22.07.2019.
OIB Pristupnika:	55478864133
Mentor:	Doc. dr. sc. Bruno Zorić
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Napadi na steganografske sustave
Znanstvena grana rada:	**Obradba informacija (zn. polje računarstvo)**
Zadatak završnog rad:	U teorijskom dijelu rada potrebno je najprije opisati pojam steganografije te različite postupke za skrivanje informacija uporabom iste. Prikazati pri tome različite primjene steganografskih sustava. Zatim je potrebno opisati moguće napade na ovakve sustave namijenjene razotkrivanju skrivene informacije. U praktičnom dijelu rada ostvariti programsko rješenje koje pruža mogućnost provođenja odabranih napada i otkrivanje informacija u porukama kreiranim uporabom steganografije. Tema rezervirana za: Gabriel Veselovac
Prijedlog ocjene završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	31.08.2022.
Datum potvrde ocjene od strane Odbora:	21.09.2022.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 21.09.2022.

Ime i prezime studenta:	Gabriel Veselovac
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R 4437, 22.07.2019.
Turnitin podudaranje [%]:	2

Ovom izjavom izjavljujem da je rad pod nazivom**: Napadi na steganografske sustave**

izrađen pod vodstvom mentora Doc. dr. sc. Bruno Zorić

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 2

2. STEGANOGRAFIJA 3

2.1. Osnovni pojmovi i slučajevi korištenja 3

2.1.1. Povijesni pregled 4

2.1.2. Mediji i tehnike 5

2.1.3. Slučajevi korištenja 6

2.2. Postupci provođenja steganografije 7

2.2.1. Metoda najmanjeg značajnog bita (LSB) 8

2.2.1. Metode temeljene na razlici vrijednosti piksela (PVD) 10

2.2.2. Metode kodiranja poruka unutar teksta 11

2.2.3. Metoda digitalne kardanske rešetke (DCG) 12

2.2.4. Metoda diskretne kosinusne transformacije (DCT) 13

2.2.5. Metoda skrivanja u diskovni prostor 15

2.2.6. Metoda skrivanja unutar internetskih mreža 15

2.2.7. Steganografija segmentacije složenosti bitnih razina (BPCS) 16

2.3. Alati za provođenje steganografskih postupaka 17

3. NAPADI NA STEGANOGRAFSKE SUSTAVE 20

3.1. Osnovni pojmovi i klasifikacija 20

3.2. Tehnike i postupci za provođenje napada 21

3.2.1. Vizualna detekcija 21

3.2.2. Statistička analiza 22

3.2.3. Strukturna detekcija 24

3.2.4. Analiza stranih kanala 26

3.2.5. Metode obasipanja 26

3.2.6. Analiza potpisa 27

4. PROGRAMSKO RJEŠENJE ZA PROVOĐENJE NAPADA NA STEGANOGRAFSKE SUSTAVE 28

4.1. Opis aplikacije i specifikacija zahtjeva 28

4.1.1. Python i KivyMD 29

4.1.1. Pillow (PIL) i Matplotlib 29

4.1.1. Multiprocessing, OpenCV (cv2) i NumPy 29

4.1.2. Pyexiv2 (exiv2) i Os 29

4.2. Dijagram toka **30**

4.3. Mogućnosti i rad aplikacije **30**

4.3.1. Korištenje i testiranje aplikacije 36

4.4. Osvrt i mogućnosti poboljšanja **42**

5. ZAKLJUČAK **43**

LITERATURA **44**

ABSTRACT **49**

ŽIVOTOPIS **50**

PRILOZI **51**

1. UVOD

Ubrzanom digitalizacijom i dostupnošću informacija javljaju se problemi njihova integriteta, sigurnosti, pristupa i autentičnosti. Razmjena i objava slika, videozapisa, tekstualnih i ostalih podataka odvija se svakog trenutka na raznim platformama i komunikacijskim kanalima. Osim dostupnih tehnologija i lakoće razmjene informacija, međusobna digitalna povezanost igra važnu ulogu u razmjeni istih – čime se otežava kontrola i nadzor. Prema [1], svake je sekunde tokom 2020. godine stvoreno 1,7 megabajta podataka, dnevno objavljeno oko 500 000 objava na Twitter-u te razmijenjeno 306,4 milijarde elektroničkih poruka. Nadalje, prema [2], 2022. godine je svake sekunde objavljeno više od 8000 poruka na platformi Snapchat te je na WhatsApp-u zabilježen broj od ukupno milijardu postojećih videozapisa te 4,5 milijarde slika. Upravo je zbog ove dostupnosti i količine informacija potrebno analizirati i utvrditi sve moguće napade i učinkovite metode za zaštitu i prevenciju [3] – čime sigurnost multimedijskih podataka postaje jedno od ključnih pitanja javnog interesa [4]. Kao rješenje javlja se šifriranje poruka, no ono privlači pažnju. Stoga je potrebna nevidljiva komunikacija koju steganografija omogućuje. No, osim osiguravanja tajnosti privatnih poruka, steganografske tehnike mogu sakriti i razne zlonamjerne programe (engl. *Stegware*) [5]. Ovi programi mogu se koristiti za razne operacije poput prodiranja u sustave, krađu informacija te okidanje sakrivenih zloćudnih programa. Prema [6], ruski hakeri infiltrirali su američki obrambeni sustav koristeći zlonamjerne programe kodirane unutar slika postavljenih na Twitter-u. Nadalje, prema [6], primijećene su brojne komunikacije omogućene uz pomoć steganografije između raznih terorističkih organizacija poput Hamas, Hezbollah i Al Qaeda. Danas je stoga potrebno osigurati (osim tajnosti) i sigurnost prikrivenih informacija. U priču se javljaju napadi na steganografske sustave. Oni obuhvaćaju otkrivanje prisutnosti steganografije te sakrivenih informacija (engl. *stegoanalysis*) [7], što će predstavljati fokus ovog rada.

U drugom poglavlju detaljno se opisuje pojam steganografije, postupaka, medija i alata provođenja. Treće poglavlje opisuje napade na steganografske sustave, primjenu i moguće alate. Četvrto poglavlje predstavlja implementaciju aplikacije koja provodi odabrane napade na steganografske sustave te testiranje i opis načina rada aplikacije. Peto poglavlje daje osvrt na ostvareno.

1.1. Zadatak završnog rada

U teorijskom dijelu rada potrebno je opisati pojam steganografije, različite postupke za skrivanje informacija uporabom iste te različite primjene steganografskih sustava. Zatim je potrebno opisati moguće napade na ovakve sustave namijenjene analizi i razotkrivanju skrivene informacije. U praktičnom dijelu rada ostvariti programsko rješenje koje pruža mogućnost provođenja odabranih napada i pokušaj otkrivanja informacija u porukama kreiranim uporabom steganografije.

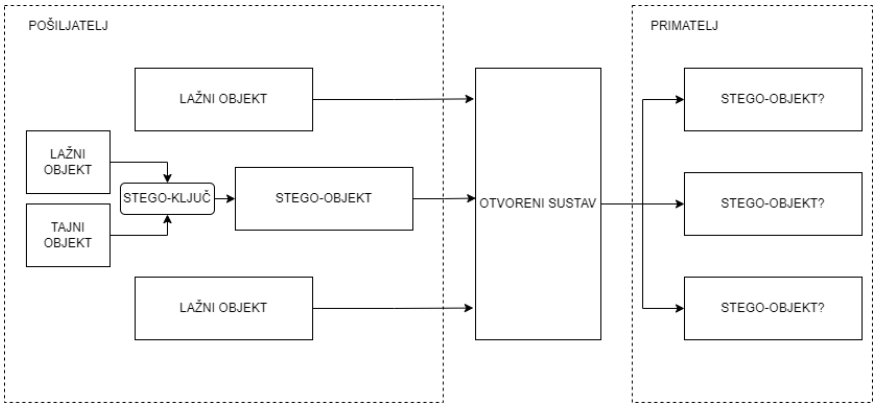
2. STEGANOGRAFIJA

Sigurnost podataka i otpornost na napade važan je aspekt današnjice. Steganografija pruža alternativu za već poznate metode kriptografije. Otvara vrata mnogim prednostima, ali i opasnostima.

2.1. Osnovni pojmovi i slučajevi korištenja

Steganografija je umjetnost i znanost nevidljive komunikacije [8]. Dolazi od grčkih riječi *steganos* („pokriven, skriven, zaštićen“) i *grafein* („pisanje“) [9, 14], u prijevodu „skriveno pisanje“. To je praksa prikrivanja tajnih poruka unutar nekog drugog podatka (medija), a istražuje kako i na koji način željene poruke sakriti. U relaciju sa steganografijom ulazi i kriptografija, koja je zasnovana na čuvanju sadržaja. Ciljevi su različiti – kriptografija pokušava onemogućiti manipuliranje ili čitanje poruke, dok se steganografija usredotočuje na čuvanje njene postojanosti.

Od važnih pojmova javljaju se: lažni objekt (lažna poruka, izvor, original; engl. *cover-object/ cover-message*), unutar kojeg se ugrađuje željena poruka, stego-objekt (engl. *stego-object*), objekt koji nastaje kada se poveže lažna i željena poruka te stego-ključ (engl. *stego-key*), koji pruža dodatne informacije potrebne za kontrolu procesa skrivanja [12, 15, 16]. Npr. slika 1.1., gdje osoba A (pošiljatelj) želi poslati osobi B (primatelj) tajnu poruku. Kako bi to učinila, osoba A ugrađuje željenu poruku u jedan od lažnih objekata (npr. odabranu sliku, video, zvukovni isječak, tekst i dr.). Dobiva se stego-objekt koji se, zajedno s drugim lažnim objektima, šalje kroz određeni otvoreni sustav (npr. Internet). Otvoreni sustav je sustav koji redovito razmjenjuje informacije s vanjskim okruženjem, a predstavlja povratni ciklus svih ulaza, procesa, izlaza, procjena i evaluacija [11]. Stego-ključ može služiti kao dodatna zaštita potrebna pri izvlačenju podataka, ili kao „tok“ za skrivanje. Nastaje problem raspoznavanja lažnih objekata od objekata koji sadrže poruku (stego-objekti), kao i proces iščitavanja skrivene poruke. U savršenom sustavu, lažan objekt bit će neprepoznatljiv u usporedbi sa stego-objektom (bilo ljudskom percepcijom ili statističkim uzorcima), no ovakav sustav nije moguće realizirati [12].



Slika 1.1. Primjer jednostavnog slanja stego-objekta kroz otvoreni sustav

Prema [13], postoje 3 glavna protokola steganografije: čista steganografija (engl. *pure steganography*), steganografija tajnog ključa (engl. *secret key steganography*) i steganografija javnog ključa (engl. *public key steganography*). Čista steganografija definirana je kao steganografski sustav koji ne zahtjeva razmjenu šifre kao što je stego-ključ. Ovaj protokol najmanje je siguran upravo zbog pretpostavke da osim pošiljatelja i primatelja nijedna druga strana neće biti svjesna postojanja poruke. Steganografija tajnog ključa obuhvaća sustav koji zahtjeva razmjenu tajnog ključa (stego-ključa) prije početka razmjene informacija (komunikacije). Ovaj ključ koristi se prilikom ugradnje i izvlačenja poruke. Zbog ove razmjene ključa sustav je podložniji presretanju te privlači veću pažnju. Prednost je što isključivo strane koje posjeduju tajni ključ mogu izdvojiti skrivenu poruku. Steganografija javnog ključa koristi javni ključ prilikom kodiranja, a tajni ključ prilikom izvlačenja informacija, što sustav čini vrlo otpornim.

2.1.1. Povijesni pregled

Ideja i praksa skrivanja informacija ima dugu povijest, uz mnoštvo metoda i varijacija. Jedan od prvih dokumenata koji opisuje steganografiju dolazi iz djela Histories grčkog povjesničara Herodota [14]. U jednoj od priča Demeratus je htio obavijestiti Spartu o planiranom napadu na Grčku. Naime, u staroj Grčkoj tekst je bio pisan na drvenim pločama prekrivenim voskom. Demeratus je na drvetu ispisao tajnu poruku te ju potom prekrio novim slojem voska. Ploče su se činile prazne i neiskorištene. Druga priča spominje tetoviranje poruke na obrijanu glavu glasnika. Nakon izrasta kose poruka bi bila neotkrivena i sigurna. Nadalje, u sigurnosnom protokolu razvijenom u Kini, pošiljatelj i primatelj imali su kopije papirnate maske s brojnim rupama na nasumičnim mjestima [15]. Pošiljatelj bi stavio svoju masku na list papira, ispisao tajnu poruku i zatim sastavio okolne lažne informacije. Jedino bi primatelj bio u mogućnosti iščitati poruku tako što bi stavio svoju masku preko dobivenog pisma. U ranom 16. stoljeću talijanski matematičar Cardan (1501. – 1576.) adaptirao je ovu tehniku danas poznatu kao Cardan Grille (kardanska

rešetka) [15]. Od ostalih povijesnih događaja istaknuo bi se i primjer korištenja nevidljive tinte koje su bile primijenjene u Prvom i Drugom svjetskom ratu [14]. Izvori su najčešće bili mlijeko, ocat, voćni sokovi i dr. supstance koje potamne kada se zagriju. Poboljšanjem tehnologije razvijene su sofisticiranije tinte koje reagiraju na razne kemikalije. Osim toga, u drugom svjetskom ratu Nijemci su izmislili Microdot tehniku [8]. Informacije, posebno fotografije, sažimale su se dok nisu bile veličine točke. Izuzetno teške za otkriti, lažne poruke poslane su zajedno s tajnim informacijama preko nesigurnih kanala. Do danas, Microdot tehnika razvila se do mjere da su godine 1999. istraživači s medicinskog fakulteta Mount Sinai u New York-u kodirali skrivenu poruku na lancu ljudskog DNA, pomoću tehnike zvane DNA steganografija [14]. Nositelj poruke može biti bilo koje sredstvo sa sposobnošću za prijenos informacija, od uključujući npr. pločica od drveta, šupljih potpetica, slika ispod poštanskih markica, fotografija ili kombinacija riječi.

Danas, steganografske tehnologije predstavljaju važan aspekt za budućnost internetske sigurnosti i privatnosti na otvorenim sustavima poput Interneta.

2.1.2. Mediji i tehnike

Digitalni kanali tj. mediji uključuju elektroničku poštu, zvučne i video zapise, prostor na disku, particije diska, mreže i slike [14]. Zbog ograničenja ljudske vizualne percepcije (engl. *human visual system*, HVS) i lakoće kodiranja (skoro bilo koji tekst, slika ili drugi tip podatka može se kodirati kao tok bitova unutar slike), slike predstavljaju jedan od najčešće korištenih medija [13]. Modificiranje slika temelji se na vrlo malim izmjenama nad pikselima unutar prostorne (engl. *spatial*) ili frekvencijske (engl. *transformed*) domene [16]. Uz to, statističke i vizualne karakteristike izvorne slike ostaju očuvane. Video poruke zahtijevaju složeniju implementaciju jer najčešće kombiniraju dvije ili više tehnika, a informacije se najčešće ubacuju unutar pojedinih video okvira [17]. Prostor na disku koristi se na temelju neiskorištenog, rezerviranog ili particioniranog prostora, čija primjena ne degradira medij kao ostali kanali. Ovaj rezervirani prostor poznat je pod nazivom *slack space* [14] (više o opisu tehnike koja iskorištava ovu slabost unutar poglavlja 2.2.5). U ostale medije ulazi tekst, no njegovo kodiranje može biti vrlo izazovno jer se temelji na suvišnim podacima kojih tekst ima vrlo malo [13]. Od problema javlja se formatiranje ili direktne promjene na sadržaj teksta.

Od tehnika steganografije ističu se injekcija, supstitucija te generiranje novih datoteka [13]. Injekcija se odnosi na umetanje poruke u postojeći medij. Supstitucija zamjenjuje određene podatke originalnog objekta sa novokodiranim reprezentacijama kombinacije originalnih podataka i podataka tajne poruke. Nadalje, injekcija i supstitucija zahtijevaju takozvanu *host* datoteku (već

spomenuti lažni-objekt; izvor, original unutar kojeg se žele pohraniti informacije). Usporedba *host* datoteke i generiranog stego-objekta može dovesti do pojave uzoraka koje steganografski analitički alati koriste kako bi detektirali prisutnost tajne poruke. Generiranje novih datoteka rješava ovaj problem tako što se vezanje podataka odvija pri stvaranju medija, pri čemu stego-objekt u jednu ruku postaje svoj original. Npr. alat *Spam Mimic* omogućuje kodiranje tajnih poruka unutar poruka koje izgledaju kao neželjene (engl. *spam*) poruke [13].

2.1.3. Slučajevi korištenja

Što se tiče slučajeva korištenja, postoje razne vrste s obzirom na tip i svrhu. Najčešći primjeri temelje se na otvorenim sustavima. Prema [12], skriveni kanali unutar TCP/IP protokola uključuju maskiranje identifikacijskih podataka na Internetu kada je potrebna potpuna tajnost. Vrlo često je i korištenje lažnih poruka unutar kojih se ugrađuju tajne poruke pri povjerljivoj, privatnoj ili vrlo osjetljivoj komunikaciji, posebice pri razmjeni dokumenata. Nadalje, vodeni žigovi (engl. *watermarking*) predstavljaju uzak dio steganografije zbog korištenih steganografskih postupaka i tehnologija [12]. Koriste se pri osiguravanja vlasništva i otkrivanju digitalnih krađa. U moderne tehnike ulazi skrivanje poruka u manipuliranim izvršnim datotekama (iskorištavanje suvišnosti u i386 skupu instrukcija), ugrađivanje slika u video datoteke (pri alterniranju brzine reprodukcije), ubrizgavanje neprimjetnih kašnjenja u pakete koji se šalju kroz mrežu (kašnjenje se može iskoristiti za određivanje koji podatak prolazi mrežom, princip kodiranja), BPCS-steganografija tj. steganografija vrlo velikog ugradbenog kapaciteta, blog-steganografija (poruke su podijeljene na fragmente, a šifrirani dijelovi su postavljeni kao komentari napuštenih web-stranica tj. blogova čiji odabir predstavlja simetrični ključ koji primatelj i pošiljatelj koriste; nositelj skrivene poruke je cijeli skup svih web-stranica) i dr. [13]. S druge strane, steganografija se koristi i u svrhe napada na sustave i korisnike. *Stegware* je pojam koji označava korištenje steganografije u svrhu prikrivenog ugrađivanja zlonamjernog softvera [5]. Javne web-stranice mogu sadržavati tajne informacije ili skriveni kod ugrađen u raznim resursima poput slika, a s obzirom da slikovni tip prometa generalno nije sumnjiv ništa se neće činiti neobičnim. Zlođudni kod tada može nanijeti puno štete, npr. skidanjem ucjenjivačkog softvera (engl. *ransomware*), napadom na operacijski sustav, krađom osobnih podatke i dr. Osim toga, na javno dostupnim slikama moguće je podijeliti razne povjerljive informacije. Primjera je mnogo, a opasnost korištenja steganografije u napadima potvrđuju i poznati antivirusni poslužitelji:

„Nažalost, korištenje steganografije unutar kibernetičkih napada lako je za implementirati i teško za otkriti."

McAfee [18]

„Danas se, međutim, pojavljuje novi opasan trend: steganografija se sve više koristi za stvaranje zlonamjernog softvera i alata za cyber špijunažu. Većina modernih rješenja protiv zlonamjernog softvera pruža malu (ili nepostojeću) zaštitu od steganografije, dok bilo koji medij može sadržavati potencijalnu prijetnju.“

Alexey Sculmin, Evgeniya Krylova [19]

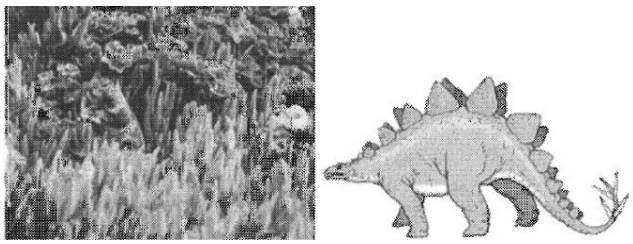
Iz navedenoga, jasna je potreba za adekvatnom analizom steganografskih sustava, obrađenom u poglavlju 3. Osim osiguravanja privatnosti korisnika mora se pružiti i zaštita protiv zloćudnog iskorištavanja ovih alata.

2.2. Postupci provođenja steganografije

Nekoliko ključnih svojstava mora se uzeti u obzir pri provođenju steganografije, odnosno izradi stego-objekta. Što se tiče kvalitete metoda, prvotno je bitna neosjetljivost (engl. *imperceptibility*), koja označava da korisnik ne bi trebao biti u mogućnosti razlikovati original (lažni objekt) od stego-objekta [20]. Nadalje, treba odrediti kapacitet ugradnje (engl. *embedding capacity*) koji se odnosi na količinu informacija koje se mogu ugraditi bez degradacije kvalitete slike [20] (objekt mora sačuvati svoje „značenje“ tj. reprezentaciju). Npr. za zvučne zapise kapacitet se mjeri po količini bitova prenesenim po sekundi (engl. *bit/ sec*, BPS), bitova po uzorku (engl. *bits/ sample*), bitova po bajtu (engl. *bits/ byte*), kvantskih bitova (engl. *qubits/ sample*), bitova po odabranom koeficijentu (engl. *bits/ sc*) ili bitova po okviru (engl. *bits/ frame*) [10]. Za digitalne slike kapacitet predstavlja prosječan broj bitova skrivenih unutar piksela lažnog objekta (engl. *bits per pixel*, BPP) ili bitova po koeficijentu (engl. *bits per coefficient*, BPC) [10]. Za zvučne zapise kapacitet se najčešće mjeri bitovima po pikselu po okviru (engl. BPP/ *frame*), a za 3D zapise bitovima po pikselu po okviru ili bitovima po vrhu (engl. *bits per vertex*, BPV) te bitovima po koordinati (engl. *bits/ coordinate*), npr. unutar x,y i z koordinatne ravnine [10]. Nadalje, bitna je robusnost (engl. *robustness*), težina potrebna za uništavanje ugrađenih informacija bez degradacije originalnog objekta [20]. Osim navedenog, postoje dodatni faktori koji se mogu iskoristiti pri što kvalitetnijoj ugradnji informacija kao što su fiziološke karakteristike promatrača [21] (npr. ljudi su relativno neosjetljivi na promjene između crvene i ljubičaste tj. imaju nižu osjetljivost na plavu boju), udio jačine signala naspram jačine šuma (engl. *Signal-To-Noise Ratio*, SNR), indeks strukturne sličnosti (engl. *Structural Similarity Index*, SSI), srednje ocjene mišljenja (engl. *Mean opinions Score*, MOS) i mnogi drugi [10].

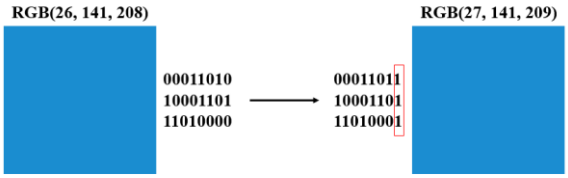
2.2.1. Metoda najmanjeg značajnog bita (LSB)

Steganografija najmanje značajnog bita (engl. *least significant bit*, LSB) jedna je od temeljnih i konvencionalnih metoda sakrivanja većih količina informacija bez primjetnih vizualnih izobličenja [22]. Radi na principu zamjene LSB-a slike s dijelovima tajne poruke, gdje se odabir piksela ili redoslijed ugradnje može odrediti stego-ključem [22] (algoritmom, „tokom“). Potrebno je razmotriti strukturu slike kao i paletu boja kako bi se što bolje odabrala područja ugradnje, tj. područja unutar kojih je teže uočiti vizualne razlike nakon ugradnje bitova [14]. Fokus se stavlja na ljudsku (ne)sposobnost razlikovanja malih promjena, tj. vizualnih karakteristika slika [9]. Slika 2.1 prikazuje dvije slike različitih tekstura, gdje svaka od njih ima područja povoljna za ugradnju. Lijeva slika puno je detaljnija, iz čega se može zaključiti da mala izmjena većine piksela neće predstavljati gotovo nikakvu vizualnu razliku. S druge strane na desnoj slici područja se moraju birati pažljivije jer će izmjene biti uočljivije.



Slika 2.1. Slike s različitim teksturama, prema [14]

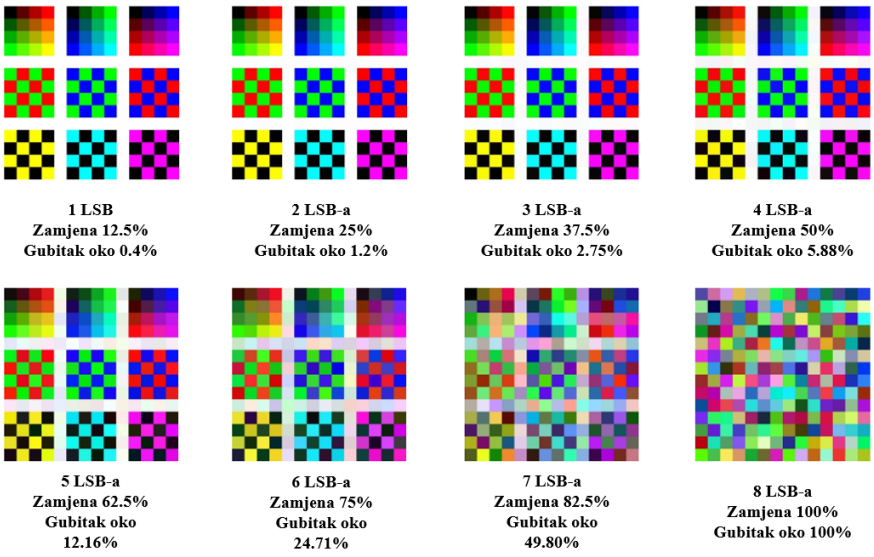
Jednostavna implementacija ove tehnike prikazana je na slici 2.2., na primjeru plavog piksela 24-bitne slike. Prijelazom RGB(26, 141, 208) reprezentacije piksela u bitni oblik dobiva se polje veličine 3x8 (gdje svaki redak predstavlja intenzitet pojedine boje: crvene, zelene, plave). Ukoliko se uzme u obzir želja da se ugradi bitna informacija „111“, reprezentacija piksela poprima vrijednost RGB(27, 141, 209). Kao što se na slici može vidjeti, razlika je neprimjetna. Isti princip primjenjuje se na veće količine informacija, izmjenom LSB-a pojedinih piksela.



Slika 2.2. Jednostavno ugrađivanje niza bitova „111“ u RGB(26,141,208) piksel

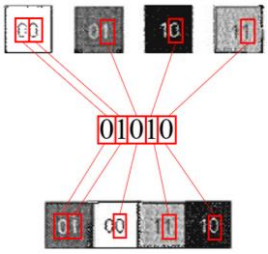
Nadalje, u obzir treba uzeti maksimalnu količinu bitova koju je moguće iskoristiti. Ovisno o tipu slike (24-bitne, 8-bitne) ovisi i utjecaj vizualne promjene. Na slici 2.3. prikazana je vizualna

razlika u ovisnosti o promjeni *n* LSB-a 24-bitne slike [9]. Npr. ukoliko se raspolaže slikom rezolucije 3840×2160 bita, u teoriji je moguće maksimalno sakriti 6220800 bita podataka (75% sveukupnog). U suprotnom dolazi do prevelike promjene izvorne slike.



Slika 2.3. Prikaz perceptivne promjene pri mijenjanju određenog broja bitova piksela 24-bitne slike, prema [9]

S druge strane, manipulacija 8-bitnih slika nije toliko fleksibilna [14]. Npr. na slici 2.4., pri ugradnji niza bitova „01010“. LSB-ovi se izmjenjuju tako da njihova kombinacija poprimi oblik željenog niza bitova tj. tajne poruke. Kao što se može vidjeti, zamjena 1 LSB-a jasno ističe razliku originala i novodobivene slike. Razlog je reprezentacija – 24-bitne slike koriste 3 bajta za prikaz 1 piksela, dok 8-bitna koristi 1 (paleta boja je toliko malena da su promjene lako vidljive) [14].



Slika 2.4. Prikaz vizualne razlike pri promjeni LSB-a 8-bitne slike, prema [14]

LSB tehnika koristi digitalne medije poput slika, zvučnih, video ili 3D zapisa [10]. Prema [13], problem ove tehnike je u mogućnosti lakih napada poput formatiranja (promjena datoteke iz gif u pdf format) ili bilo kojih drugih direktnih alternacija. Nadalje, kao što je vidljivo iz [10], šum generiran uz korištenje LSB metode pruža laku primjenu steganografsko-analitičkih metoda za dokaz o postojanju skrivenih podataka (više u poglavlju 3). Kao posljedica toga, često je korištena uz kombinaciji s kriptografijom.

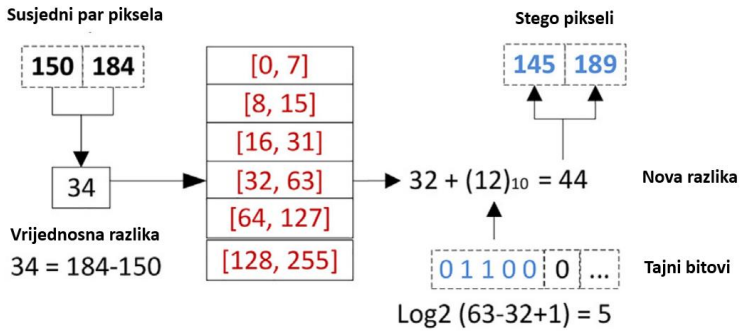
2.2.1. Metode temeljene na razlici vrijednosti piksela (PVD)

Iako su LSB metode vrlo jednostavne i fleksibilne za korištenje, broj izmijenjenih LSB-ova uvelike utječe na vizualnu kvalitetu stego-objekta [22]. Kod metoda temeljenih na razlici vrijednosti piksela (engl. *pixel value difference*, PVD) lažan objekt dijeli se na *n* susjednih blokova od dva piksela koji se ne smiju preklapati s drugim susjednim blokovima [23]. Ovo označava da je svaki piksel unikatno prisutan unutar svog para, nijednog drugog. Razlika vrijednosti između dva susjedna piksela određuje koliko tajnih bitova je moguće ugraditi – veća razlika, veći ugradbeni kapacitet [22]. Vrijednost pojedinog piksela označava broj između [0, 255], tj. predstavlja intenzitet sivog piksela ili svjetlosne komponente, npr. Y komponenta YCbCr područja (više o YCbCr području u poglavlju 2.2.4).

Npr. na slici 2.5. Za svaki blok piksela izračunata je vrijednosna razlika. Ta razlika korigira se na temelju tablice gdje su vrijednosne razlike grupirane u raspone, čiji se odabir temelji na osjetljivosti ljudskog vida [22]. U ovom primjeru razlika susjednih bitova (150, 184) iznosi 34, što spada u interval [32, 63]. Nakon korigiranja razlika prelazi u vrijednost 32, te zajedno s izračunatom količinom tajnim bitovima (5 bitova, vrijednost „01100“) čini novu razliku vrijednosti 44. Količina tajnih bitova koji se mogu ugraditi u dani par računa se formulom:

$$log_2(gg - dg + 1) \tag{2-1}$$

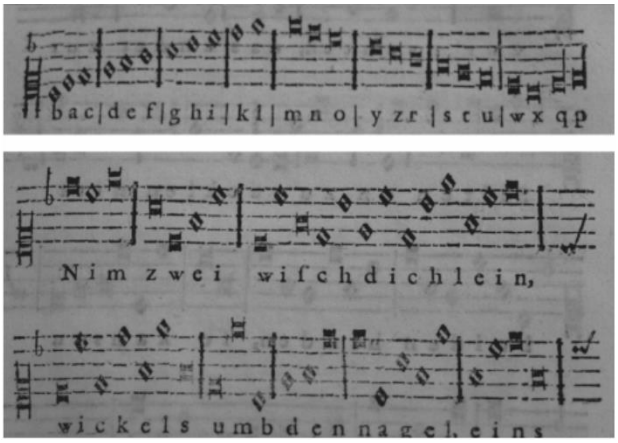
gdje *gg* označava gornju granicu odabranog raspona, a *dg* donju granicu odabranog raspona [22]. Nova razlika i broj tajnih bitova koriste se pri izmjeni susjednog para piksela kako bi se dobili novi stego-pikseli, te se postupak ponavlja sve dok se ne ugradi cijeli niz tajnih bitova.



Slika 2.5. Princip rada PVD tehnike, prema [22]

2.2.2. Metode kodiranja poruka unutar teksta

Kodiranje poruka unutar teksta može biti vrlo izazovan posao, a problem je u malenoj količini suvišnih podataka, koji su potrebni za skrivanje tajnih poruka [13]. Za razliku od slike, čijom izmjenom dolazi do teže vidljivih promjena, tekst označava potpunu reprezentaciju željene informacije. Kao što je uočljivo na prijašnjim primjerima, male promjene nad 24-bitnim slikama ne predstavljaju drugo „značenje“. Stoga je pitanje kako promijeniti npr. riječ „steganografija“ da i dalje znači „steganografija“. Promjena jednog bita u reprezentaciji znači potpunu promjenu slova, što nije pouzdana tehnika. Osim toga, steganografske tehnike vezane uz tekst vrlo su podložne i osjetljive na promjene poput formatiranja ili izravne alternacije [13], što povlači da je u većini slučajeva potrebna originalna datoteka. Zbog navedenog nedostatka u količini suvišnih podataka, metode kodiranja poruka unutar teksta uglavnom se realiziraju manipulacijom linija i riječi [14]. Npr. Gaspar Schott (1608.-1666.) primijenio je varijantu ove tehnike tako što je kodirao note sa slovima engleske abecede [15] (slika 2.6.).



Slika 2.6. Steganografska tehnika skrivanja informacija u notni zapis (kodiranje poruke unutar teksta), prema [15]

Nadalje, prema [13], kodiranja poruka unutar teksta dijele se na kodiranja s pomakom linija (engl. *line-shift encoding*), kodiranja s pomakom riječi (engl. *word-shift encoding*) te kodiranja specifičnih značajki (engl. *feature specific encoding*). Kodiranje s pomakom linija predstavlja pomicanje svake linije teksta vertikalno gore ili dolje za duljinu od 3 centimetara (ili manje). Pomaci gore ili dolje lako se mogu supstituirati za bitne vrijednosti 0 ili 1, koje dalje mogu predstaviti bilo koju pisanu informaciju. Kodiranje s pomakom riječi funkcionira na isti način, samo što se koriste horizontalni razmaci. Kodiranje specifičnih značajki uključuje promjenu atributa teksta poput vertikalne ili horizontalne duljine slova kao što su „b“, „d“, „T“ i dr. [13]. Nadalje, HTML datoteke mogu se koristiti za dodavanje razmaka, tabulatora, „nevidljivih“

znakova i dodatnih redaka koje web preglednici zanemaruju. Dodatni razmaci nisu vidljivi dok se ne otkrije izvor web stranice.

2.2.3. Metoda digitalne kardanske rešetke (DCG)

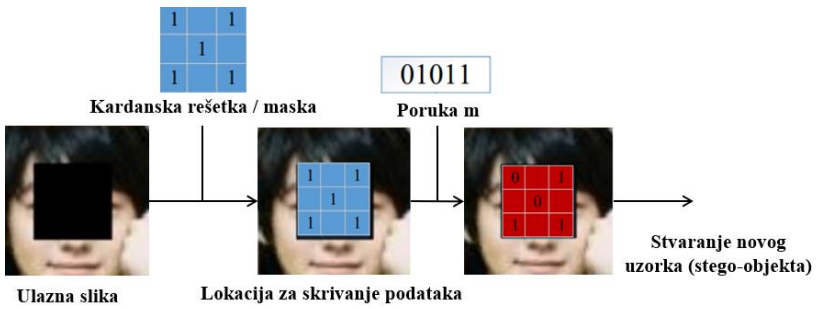
Digitalna kardanska rešetka (engl. *Digital Cardan Grille*, DCG) predstavlja proširenje na već poznatu klasičnu tehniku kardanske rešetke, a označava masku za određivanje skrivene poruke unutar digitalnih medija [24]. To je postupak stvaranja relacijskog odnosa između tajne poruke i stego-objekta, odnosno preslikavanje i izdvajanje skrivenih podataka. Npr. slika 2.7. – prvotno se stvara oštećena regija te maska (DCG) koja će odrediti gdje će se podaci unutar te oštećene regije ugraditi. Ta maska M_{CG} (matrična vrijednost kardanske rešetke) predstavlja određenu binarnu matricu. Npr. za masku:

$$M_{CG} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \tag{2-2}$$

gdje vrijednosti 0 i 1 određuju dijelove unutar kojih se želi tj. ne želi sakriti željena poruka *m* [24]. Primjenjujući danu masku M_{CG} nad binarnom porukom *m* vrijednosti „01011“ dobiva se nova binarna poruka *m'*, koja iznosi

$$m' = \begin{bmatrix} 0 & & 1 \\ & 0 & \\ 1 & & 1 \end{bmatrix}, \tag{2-3}$$

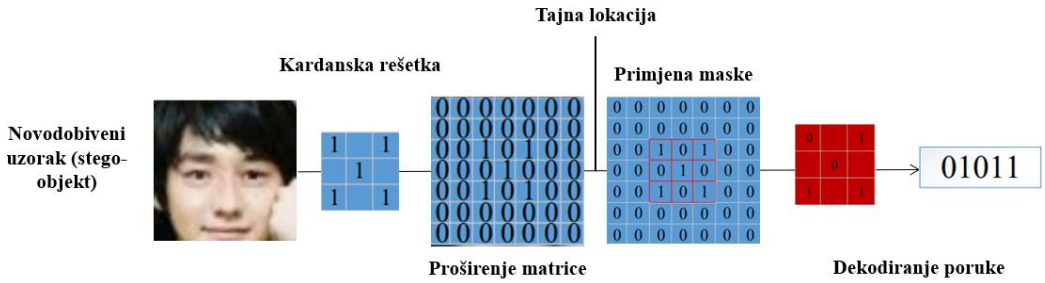
a prazne regije sadržavati će ostale bitove slike jednake bitovima regije koju maska pokriva [24].



Slika 2.7. Primjer procesa stvaranja stego-objekta uz pomoć digitalne kardanske rešetke, prema [24]

Kardanska rešetka mijenja bitove naznačene unutar definirane maske. Nakon stvaranja novog uzorka (stego-objekta) primatelj koristi istu masku kako bi odgonetnuo tajnu poruku, čiji je proces prikazan na slici 2.8. Maska se primjenjuje direktno na novi stego-objekt uz potrebna proširenja kako bi joj veličina odgovarala veličini stego-objekta, a u suprotnom dolazi do ne podudaranja

lokacije. Podudaranje bitova slike (crvena označena matrica sa slike 2.8.) sa danom maskom predstavlja tajnu poruku *m*.



Slika 2.8. Ekstrakcija tajnih podataka, prema [24]

Postoje razne tehnike ugradnje, a danas su najviše vezane uz generativne suparničke mreže (engl. *Generative Adversarial Networks*, GAN) [24], predstavljene 2014. godine [25]. Steganografski cilj GAN-a je procijeniti potencijalnu distribuciju željenih informacija te generirati nove uzorke odabrane distribucije [24].

2.2.4. Metoda diskretne kosinusne transformacije (DCT)

Diskretna kosinusna transformacija (engl. *discrete cosine transform*, DCT) koristi se unutar jpeg formata, a služi pri predstavljanju nizova podataka kao sume kosinusnih valova [14, 15]. Steganografska metoda temeljena na DCT-u zasnovana je na promjeni LSB-a DCT koeficijenata različitih od nule [8, 15]. Kako se kodiranje odvija unutar frekvencijske domene, promjene neće biti vidljive golim okom. Pri primjeni nad slikom podijeljenom na blokove veličine 8x8, DCT proizvodi 64 koeficijenata unutar čijih se kvantiziranih najmanje značajnih bitova mogu ugraditi željeni podaci [7].

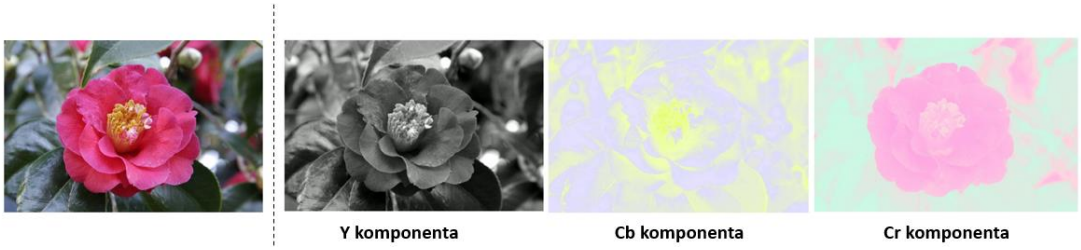
Prvotno je potrebno obaviti prijelaz iz RGB u YCbCr prostor boja, a transformacija se odvija uz pomoć sljedećih relacija:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.279 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{2-4}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0.000 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0.000 \end{bmatrix} \begin{bmatrix} Y - 16 \\ C_b - 128 \\ C_r - 128 \end{bmatrix}, \tag{2-5}$$

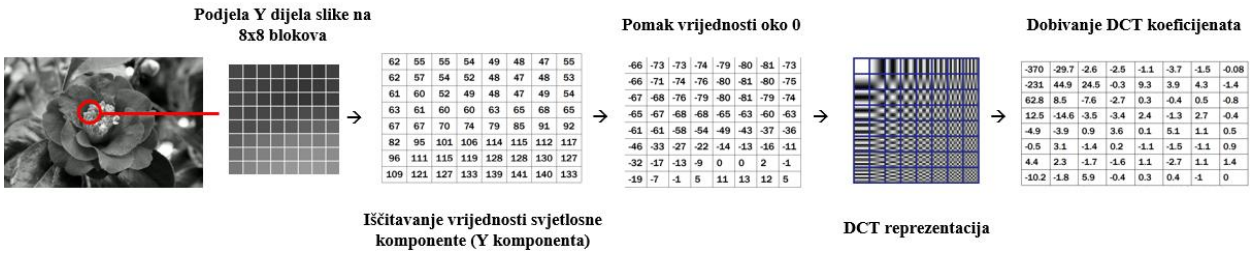
gdje vrijednosti R, G, B te Y, C_b i C_r predstavljaju doprinos pojedine komponente YCbCr prostora [26]. Za razliku od RGB prostora boja, gdje je svaki piksel predstavljen udjelom plave, crvene i

zelene komponente, YCbCr predstavlja piksel preko svjetlosne komponente (engl. *luma component/ light intensity*, Y), Cb komponente (engl. *chroma blue*, Cb) te Cr komponente (engl. *chroma red*, Cr). Cb i Cr komponente vezane su uz zelenu komponentu, a njihove relacije čine potpunu reprezentaciju slike [27]. YCbCr prostor je pogodan jer oponaša ljudski vid, tj. iskorištava činjenicu da je ljudsko oko osjetljivije na promjene intenziteta svjetlosti, a manje na promjene nijansi [26]. Kada postoji slika prikazana unutar YCbCr reprezentacije (slika 2.9.), potrebno je uzeti jedan dio te slike veličine 8x8 piksela te iščitati vrijednosti Y komponente (slika 2.10.). Nakon toga te se vrijednosti translatiraju oko nule, tj. približe kosinusnoj reprezentaciji, gdje valovi osciliraju oko nule.



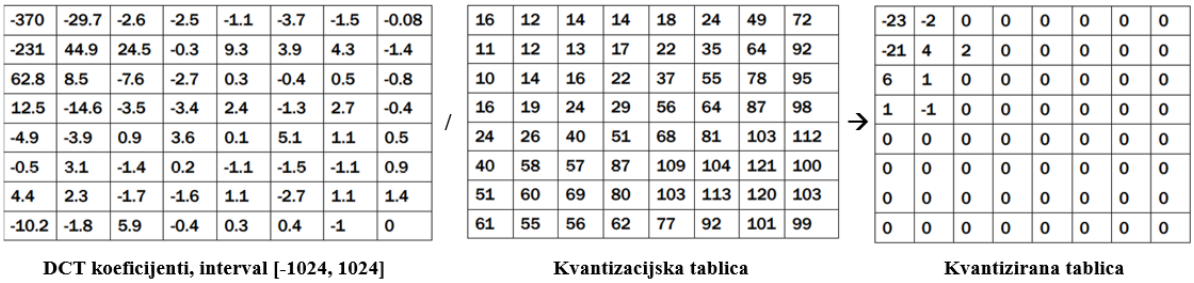
Slika 2.9. Reprezentacije slike preko YCbCr prostora boja, prema [9]

Nakon primjene DCT-a dobiva se 64 DCT koeficijenata, čija vrijednost u intervalu [-1024, 1024] označava doprinos pojedine komponente DCT reprezentacije u prikazu odabranog 8x8 područja (tj. utjecaj se mjeri DCT koeficijentima).



Slika 2.10. Postupak dobivanja DCT koeficijenata, prema [9]

Nakon toga odvija se kvantizacija koja uklanja manje DCT koeficijente kako bi smanjila ukupnu veličinu prostora koji slika zauzima [9]. Mogu se ukloniti i dijelovi koje ljudsko oko ne vidi, a to su valovi velikih frekvencija (slika 2.10., sličica naziva „DCT reprezentacija“). Kako bi se ovo postiglo, DCT koeficijenti dijele se sa kvantizacijskom tablicom te zaokružuju na najmanji cijeli broj [9], čiji je postupak prikazan na slici 2.11.



Slika 2.11. Proces kvantizacije DCT koeficijenata, prema [9]

DCT steganografija mijenja LSB-ove dobivenih kvantiziranih DCT koeficijenata (u pravilu različitih od nula i jedinica te onih koji se ne nalaze u gornjem lijevom kutu krajnje dobivene „matrice“) [9]. Ublažava nedostatak LSB metode pri odabiru povoljnih piksela. Iako vrlo otporna na steganografske napade, moguće ju je otkriti statističkim napadima (više u poglavlju 3).

2.2.5. Metoda skrivanja u diskovni prostor

Neiskorišten ili rezerviran prostor može poslužiti za držanje tajnih informacija bez degradacije nositelja tj. medija. Način na koji operativni sustavi pohranjuju podatke rezultira neiskorištenim prostorom koji se datotekama čini iskorišten. Ovaj „iskorišteni“ prostor naziva se *slack space* [14]. Npr. kod formatiranja diska kao FAT16 unutar Windows 95 operativnog sustava stvaraju se *cluster* područja veličine 32 kilobajta (KB) [14]. Iz ovoga slijedi da je minimalan prostor za pohranu datoteke 32 KB, što znači da datoteke manje od 32 KB stvaraju „neiskorišten“ prostor pogodan za skrivanje tajnih podataka. Također je moguće stvarati tajne particije, no one se mogu lako vidjeti prilikom pokretanja alata za konfiguraciju diska (engl. *disk configuration utility*) [28]. Drugi način skrivanja temelji se na iskorištavanju *DupeFile* slabosti integriteta [28]. On se odnosi na skrivanje datoteke istog imena i ekstenzije kao i druge lažne datoteke unutar iste hijerarhijske razine (putanje) bez prepisivanja lažne datoteke s novom datotekom. Unutar običnog rada duplikati datoteka mogu postojati ali isključivo uz različito ime ili putanju (tj. različitu hijerarhijsku razinu), a inače se javlja greška unutar operacijskog sustava da datoteka već postoji. Zasnovan je na iskorištavanju spomenutog *cluster* prostora prilikom spremanja kako bi obje datoteke bile spremljene na istom „mjestu“ [28].

2.2.6. Metoda skrivanja unutar internetskih mreža

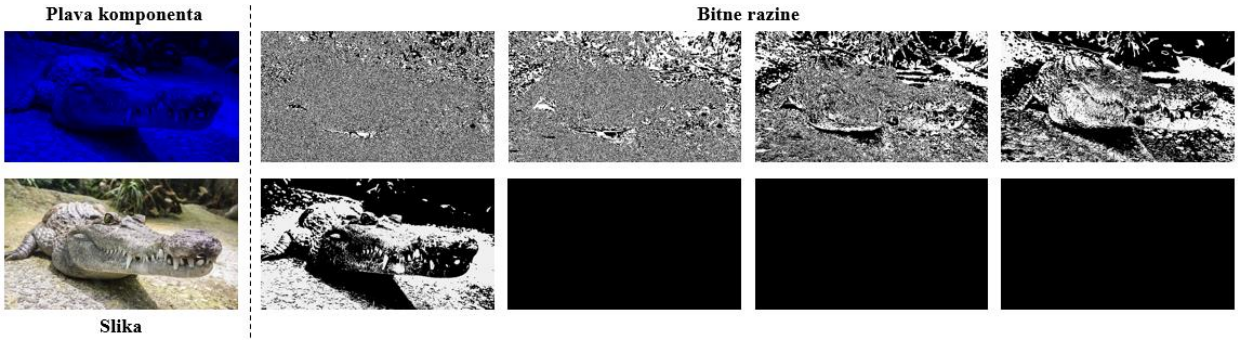
Metode skrivanja podataka unutar mreža iskorištavaju strukturu i protokole internetske mreže, kao i odvijanje razmjene željenih paketa [29]. Očituju se putem namjernih zakašnjenja i grešaka, manipulacijom funkcija definiranih za razmjenu podataka (engl. *file transfer, query-response* i sl.) te izmjenom načina rada fragmentacije i segmentacije poslanih poruka [29].

2.2.7. Steganografija segmentacije složenosti bitnih razina (BPCS)

Steganografija segmentacije složenosti bitnih razina (engl. *bit-plane complexity segmentation*, BPCS) uvedena je kao moguće rješenje na nedostatke postojećih tehnika (poput LSB tehnika, maskiranja i dr. [30]). U BPCS steganografiji originalni objekt dijeli se na regije unutar kojih se tajni podaci mogu sakriti. Za razliku od LSB tehnika, podaci se mogu sakriti unutar cijele razine tj. regije, umjesto da su te regije uvelike ograničene na pažljivo odabrane LSB bitove [30]. BPCS steganografija razdvaja sliku na skupove od *n* binarnih slika, od kojih svaka predstavlja prostor za skrivanje podataka. Tada se reprezentacija slike opisuje čistim binarnim kodom, kojim je lakše manipulirati te tako i utjecati na šum. Npr. neka postoji slika dimenzije 3x3. Neka je isto tako paleta definirana za opis te slike ograničena na 8 piksela, tj. intervalom vrijednosti $[0, 2^3 - 1]$. Na slici 2.12. prikazana je podjela ove dane slike na *n* bitnih razina (engl. *bit planes*).

Slika 2.12. Podjela 3x3 slike na bitne razine

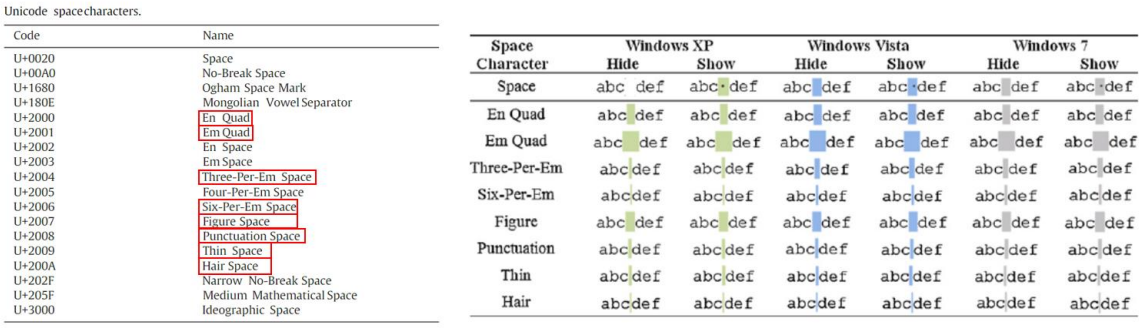
Kao što se može vidjeti, broj bitnih razina definiran je veličinom palete, tj. brojem bitova koji su potrebni kako bi se opisala svaka vrijednost te palete (u ovom slučaju to je 3 bita jer u potpunosti opisuje cijeli interval od 0 do 7). Za klasične RGB slike stoga slijedi da će postojati 8 bitnih razina ($[0, 2^3 - 1]$), za svaku komponentu zasebno (crvena, zelena, plava). Time se povećava i izbor prostora za skrivanje informacija. Primjer podjele jedne komponente slike na njene bitne razine dan je na slici 2.13. Osim podjele na bitne razine crvene, zelene i plave komponente moguće je sliku pretvoriti i u prostor sive palete te dobiti 8 bitnih razina (umjesto sveukupno 8 puta 3 za svaku pojedinu komponentu).



Slika 2.13. Podjela plave komponente slike na bitne razine

2.3. Alati za provođenje steganografskih postupaka

UniSpaCh je alat autora Lip Yee Por, KokSheik Wong i Kok Onn Chee [31]. Razmatra mješavinu razmaka između pojedinih riječi te odlomaka kako bi se podaci ugradili unutar Microsoft Word dokumenta. Radi što veće učinkovitosti i otpora na napade, koristi različite tipove UNICODE razmaka, prikazanim na slici 2.14. Kao što se može vidjeti iz [31], testiranje na različitim inačicama Microsoft Word-a pokazalo je 8 učinkovitih tipova od ukupno 18.



Unicode space characters.

Code	Name
U+0020	Space
U+00A0	No-Break Space
U+1680	Ogham Space Mark
U+180E	Mongolian Vowel Separator
U+2000	En Quad
U+2001	Em Quad
U+2002	En Space
U+2003	Em Space
U+2004	Three-Per-Em Space
U+2005	Four-Per-Em Space
U+2006	Six-Per-Em Space
U+2007	Figure Space
U+2008	Punctuation Space
U+2009	Thin Space
U+200A	Hair Space
U+202F	Narrow No-Break Space
U+205F	Medium Mathematical Space
U+3000	Ideographic Space

Space Character	Windows XP		Windows Vista		Windows 7	
	Hide	Show	Hide	Show	Hide	Show
Space	abc def	abc·def	abc def	abc·def	abc def	abc·def
En Quad	abc def	abc def	abc def	abc def	abc def	abc def
Em Quad	abc def	abc def	abc def	abc def	abc def	abc def
Three-Per-Em	abc def	abc def	abc def	abc def	abc def	abc def
Six-Per-Em	abcdef	abcdef	abcdef	abcdef	abcdef	abcdef
Figure	abc def	abc def	abc def	abc def	abc def	abc def
Punctuation	abc def	abc def	abc def	abc def	abc def	abc def
Thin	abc def	abc def	abc def	abc def	abc def	abc def
Hair	abc def	abc def	abc def	abc def	abc def	abc def

Slika 2.14. Tipovi *whitespace* znakova, prema [31]

Slika 2.15. prikazuje jedan od izlaza alata *UniSpaCh*.

Slika 2.15. Izlaz pri korištenju alata *UniSpaCh*, prema [31]

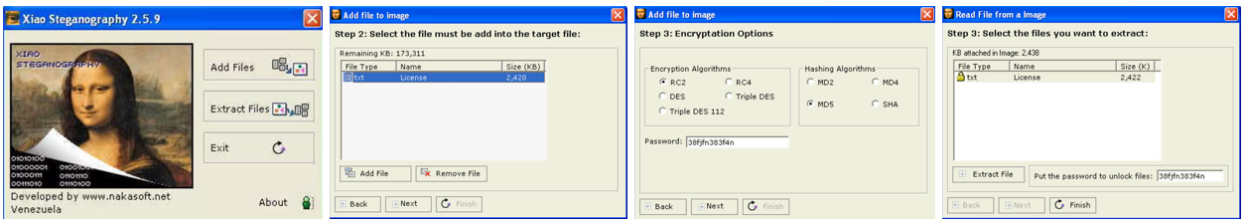
DupeFile Creator je program autora Avinash Srinivasan, Satish Kolli i Jie Wu, a iskorištava *DupeFile* slabost integriteta (poglavlje 2.2.5) [28]. Radi na jednostavnom principu odabira željene i lažne datoteke. Odabir lažne datoteke unutar mape nudi program na temelju veličine i karakteristika tajne datoteke. Program preporučuje najprikladnijih 5 datoteka, a nakon što korisnik odabere željeni objekt program napravi sve potrebne izmjene unutar trenutnog direktorija. Kao što je vidljivo i iz [28], slika 2.16. prikazuje ponašanja korištenih operativnih (engl. *operating system*, OS) i datotečnih (engl. *File Sys*) sustava. T označava *terminal*, E *explorer*, a F *finder* način rada. Y i N oznake naznačuju prikaže li operativni sustav datoteku kao duplikat datoteke ili ne, a NA označava da program nije primjenjiv (engl. *not applicable*).

OS/File Sys	FAT		NTFS		HFS+		HFS+J	
	T	E	T	E	T	F	T	F
Win 95	Y	Y	NA	NA	NA	NA	NA	NA
Win 98	Y	Y	NA	NA	NA	NA	NA	NA
Win NT	Y	Y	NA	NA	NA	NA	NA	NA
Win XP	Y	Y	Y	N	NA	NA	NA	NA
Win Vista	Y	Y	Y	N	NA	NA	NA	NA
Win 7	Y	Y	Y	N	NA	NA	NA	NA
MAC OS X	Y	N	Y	N	Y	N	Y	N

Slika 2.16. Ponašanje operativnih sustava uz različite datotečne formate pri radu *DupeFile Creator* programa, prema [28]

Dodatno, isti autori omogućili su preko *DupeFile Detector-a* te *DupeFile Extractor-a* vraćanje skrivenih podataka ugrađenih pomoću *DupeFile Creator-a* [28].

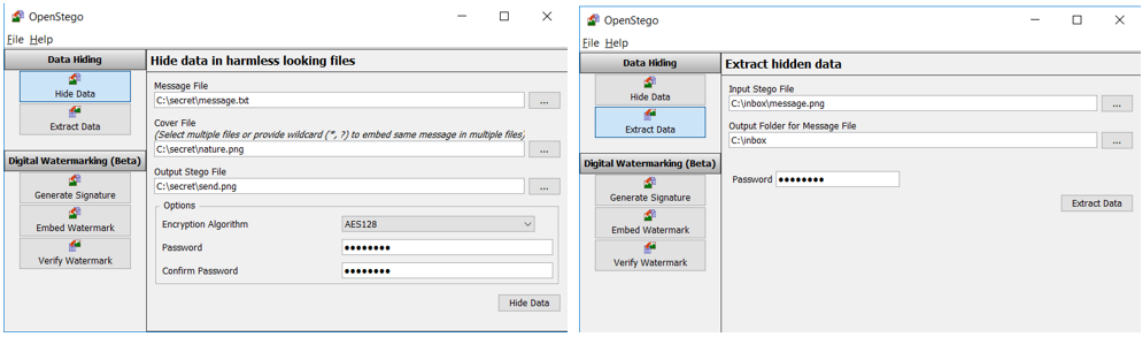
SNOW DOS je alat iz 2006. godine temeljen na metodi tekstualne steganografije, autora Kwan [31]. Pri kodiranju iskorištava razmake (engl. *whitespaces*) koji nisu vidljivi unutar većine programa za pregled teksta [32]. *WbStego4.3open* je također alat temeljen na metodama tekstualne steganografije, a koristi heksadecimalne vrijednosti „0x00“ za kodiranje „1“ i „0x20“ za kodiranje „0“ pri unosu *whitespace* znakova [31]. *S-Tools* podržava gif, mbp i wav formate te omogućuje skrivanje unutar „neiskorištenog“ prostora disketa (engl. *floppy disk*) [14] (princip iskorištavanja diskovnog formatiranja, poglavlje 2.2.5) Podržava 24-bitne slike i odabir enkripcijskih algoritama. Osim toga, pruža uvid u maksimalan moguć kapacitet, što je korisno pri odabiru medija i veličine tajne poruke. *StegoDos*, poznat još kao *Black Wolf's Picture Encoder version 0.90a*, besplatno je dostupan alat nepoznatog autora (*Black Wolf*) [14]. Sastoji se od serije DOS (*Disk Operating System*) programa, a radi isključivo na slikama dimenzija 320x200 uz paletu od 256 boja. Za kodiranje koristi LSB steganografsku tehniku. *Xiao Steganography* je besplatni softver koji se koristi za skrivanje tajnih podataka u bmp ili wav datoteke, a uz to podržava i enkripciju [33]. Prikaz rada je na slici 2.17. Alat omogućuje kodiranje i dekodiranje poruke. Prvi je korak odabir opcije, drugi odabir tajne datoteke, a treći odabir enkripcijskog algoritma, *hash* metode i lozinke.



Slika 2.17. Alat Xiao Steganography, prema [34]

Camouflage podržava bilo koje datoteke i medij ugradnje te omogućava zaštitu uz lozinku [33]. *OpenStego* podržava bmp, gif, jpeg, jpg, png i wbmp formate [33], a omogućava ugradnju

podataka i digitalnih žigova [35], ekstrakciju i ugradnju podataka te zaštitu uz lozinku [36]. Sučelje alata prikazano je na slici 2.18.



Slika 2.18. Alat OpenStego, prema [35]

SteganPEG omogućuje skrivanje bilo koje datoteke unutar slika jpg formata [33]. Omogućuje kriptiranje uz pomoć lozinki, ugradnju više datoteka unutar iste slike, računanje dostupnog kapaciteta ugradnje, sažimanje tajnih podataka prije ugradnje te selektivno dekodiranje podataka [37]. *Hide'N'Send* je maleni uslužni program koji omogućuje sakrivanje bilo koje datoteke unutar slike jpg formata te podržava haširanje i enkripciju [33]. Sučelje je jednostavno i sastoji se od dva dijela – ugradnje i izdvajanja podataka. Koristi moderne steganografske algoritme poput npr. algoritma F5. Podržava enkripcijske algoritme te *hash* funkcije [38].

Od ostalih dostupnih alata ističu se *Image Steganography*, *Steghide*, *Crypture*, *SteganographX Plus*, *rSteg*, *SSuite Picsel*, *Our Secret* [33], *Shusssh*!, *OpenPuff*, *SilentEye*, *QuickStego*, *Steganofile*, *Clotho*, *Anubis*, *Steg* [36] te mnogi drugi.

3. NAPADI NA STEGANOGRAFSKE SUSTAVE

Napade je potrebno provesti kako bi se osigurala vjerodostojnost korištene metode, ili radi uništenja zloćudnog sadržaja. Vezani su uz ciljeve napadača te korištene steganografske tehnike, tj. alate.

3.1. Osnovni pojmovi i klasifikacija

Napadi na steganografske sustave obuhvaćaju provođenje stegoanalize te uklanjanje potencijalne zlouporable steganografskih sustava. Stegoanaliza uključuje detekciju upotrebe steganografije te dekodiranje tajnih informacija (odnosno korištenih algoritama i ugrađene poruke) [4, 39]. Prema [14, 40], klasifikacija napada zasnovana je na karakteristikama i dostupnim steganografskim komponentama (tj. dostupnim informacijama), a u nju spadaju napadi preko stego-objekta (engl. *stego-only*), napadi poznatog nositelja (engl. *known-cover/ known-carrier*), napadi poznate poruke (engl. *known-message*), napadi izabrane tehnike (engl. *chosen-stego*), napadi izabrane poruke (*chosen-message*) te napadi poznate tehnike (engl. *known-stego/ known-steganography*). Napadi preko stego-objekta označavaju da je stego-objekt dostupan za analizu, dok napadi poznatog nositelja označavaju da su dostupni i original (lažni objekt) i stego-objekt. Napadi poznate poruke temelje se na analizi stego-objekata uz prvotno poznavanje tajne poruke, a svrha je otkrivanje spoznaja o korištenom sustavu i tehnici kako bi budući napadi bili uspješniji. Napadi izabrane tehnike označavaju da su algoritam/ alat te stego-objekt poznati. Kod napada izabrane poruke stego-analitičar na temelju odabranog alata/ algoritma i tajne poruke stvara stego-objekt kojeg analizira i pokušava utvrditi uzorke koji su usko vezani za te specifične algoritme tj. alate. Kod napada poznate tehnike poznati su korišten algoritam/ alat, lažni objekt (original) te stego-objekt. Poželjnije je imati što više dostupnih komponenata za analizu, no uspješnost svakog pristupa varira ovisno o korištenoj tehnici. Prema [14], jako je teško provesti napade ukoliko prvotno nije poznata tehnika ili alat. Osim toga, prema [40], napadi se mogu svrstati i u napade uništenja (engl. *destroy everything*), gdje je svrha uništiti tajnu poruku (ostalo nije bitno), napade nasumičnih karakteristika (engl. *random tweaking*), koji se temelje na dodavanju izmjena na stego-objekt kako bi tajna poruka bila nečitljiva/ nemoguća za otkriti, napade dodavanja informacija (engl. *add new information*), gdje se postojeća tajna poruka prepisuje novom raznim dodavanjima, napade reformatiranja (engl. *reformat*), gdje se odvija mijenjanje tipa datoteke te napade sažimanja (engl. *compression*), gdje se sažimanjem uvelike utječe na strukturu datoteke, što doprinosi oštećenju integriteta skrivene poruke, a time i njenog sadržaja. Osim toga, napadi se temelje i na detekciji i distorziji [14]. Detekcija zahtjeva opažanja raznih odnosa između svih procesa steganografije –

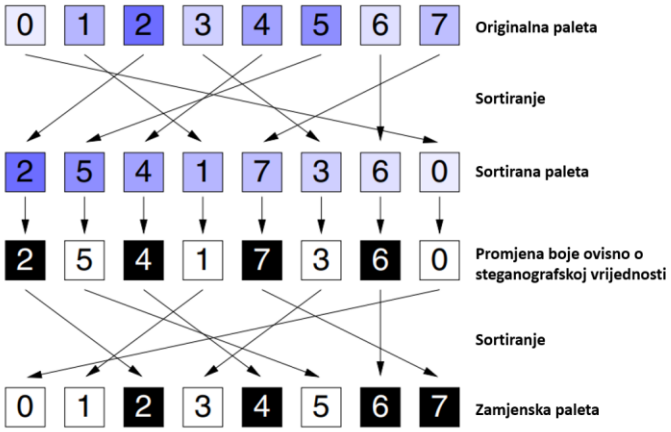
tipa stego-objekta, tipa lažnog objekta, korištene tehnike/ alata i dr. [14]. Distorzija označava manipulaciju stego-objekta do mjere uklanjanja svih skrivenih podataka. Ove radnje opažanja i manipulacije klasificiraju prirodu napada na pasivne i aktivne [14]. Što se tiče tehnika, prvi fokus stegoanalize je unutar statističke analize, tj. u statističkoj anomaliji nositelja podataka nastaloj ugradnjom skrivenih podataka [4]. Stego-objekt bi trebao imati što sličnije ili pak iste statističke karakteristike kao i original (medij/ lažni objekt) kako bi se napad što teže proveo [41]. Drugi aspekt temeljen je na stego-ključevima [41]. Kako većina tehnika steganografije koristi određeni „tok“ (algoritam, način ugradnje) potreban je stego-ključ koji služi kao posrednik između ugradnje i iščitavanja skrivene poruke. Otkrivanje ovog ključa omogućuje lako iščitavanje tajne poruke. Nadalje, za uspješno provedene napade preporučene su i druge metode poput analize stranih kanala (engl. *side channel stegoanalysis*), koja je zasnovana na otkrivanju steganografa, tj. subjekta koji provodi steganografske postupke [6]. Prema [7], stegoanaliza se primjenjuje u kibernetičkoj forenzici, kibernetičkom ratovanju, praćenju kriminalnih aktivnosti, prikupljanju dokaza za istragu, provođenju zakona, poboljšanju sigurnošću, procjeni slabosti i dr. Kako su digitalne slike kao medij za stegoanalizu najviše korištene, napadi unutar praktičnog dijela će se na njima i temeljiti. Navedene i najčešće korištene tehnike te postupci opisani su u nastavku.

3.2. Tehnike i postupci za provođenje napada

Tehnika za provođenje ima mnogo, no nijedna nije praktična u svakom slučaju. Prilikom odabira treba biti sistematičan i pažljiv – znati koje karakteristike promatrati, ili barem otkud krenuti.

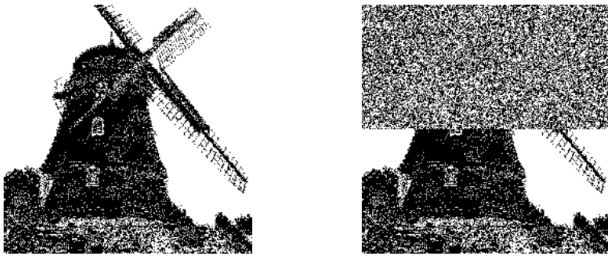
3.2.1. Vizualna detekcija

Izvorna datoteka i stego-objekti uspoređuju se za bilo koju vizualnu razliku prepoznatljivu golim okom [21, 42]. Kao što se iz prijašnjih tehnika/ alata moglo vidjeti, svaki od njih doprinosi određenim izobličenjima ili uzorcima koji se mogu promatrati. Cilj vizualnih napada je primijetiti odnos navedenih vizualnih karakteristika ili ukloniti dijelove slike koji skrivaju tajnu poruku, primjenjujući npr. razne filtere i manipulacije. Andreas Westfeld i Andreas Pfitzmann proveli su filter na LSB alatu *EzStego* koji mijenja paletu punih boja u paletu crne i bijele [41]. Rad filtera za ograničenu paletu prikazan je na slici 3.1. Originalna paleta sortira se na temelju vizualne sličnosti, promjena boje ovisi o parnosti sortirane palete, a konačno sortiranje prema indeksu stvara zamjensku paletu.



Slika 3.1. Rad filtera, prema [41]

Slika 3.2. predstavlja filtrirani rezultat originala i stego-objekta.



Slika 3.2. Proveden filter nad originalnom slikom (lijevo) i stego-objektom (desno), prema [41]

Sličan napad može se provesti i računski. Npr. jedan način detekcije uzoraka ostvaruje se sortiranjem piksela po osvjetljenju (engl. *luminance*), prema formuli:

$$Osvjetljenje = 0.299\ Crvena + 0.587\ Zelena + 0.114\ Plava \tag{3-1}$$

gdje vrijednosti *Crvena*, *Zelena* i *Plava* predstavljaju bitnu vrijednost RGB reprezentacije piksela, a *Osvjetljenje* novu vrijednost piksela [14]. Puno je lakše primijetiti uzorke unutar manjih paleta kao što su palete sivih nijansi (engl. *grayscale*) ili unutar 8-bitnih slika.

3.2.2. Statistička analiza

Statističke karakteristike slike mijenjaju se nakon ugradnje novih informacija [39]. Statistička analiza (engl. *statistical stegoanalysis*) zasnovana je na usporedbi temeljnih i novodobivenih matematičkih karakteristika, čija razlika govori je li i u kojoj mjeri datoteka promijenjena [39]. Omogućava puno bolju detekciju zbog veće osjetljivosti matematičkih tehnika naspram vizualne percepcije [7]. Dijeli se na dva tipa – specifična i univerzalna statistička analiza [39]. Specifična

analiza posebno je vezana uz korištenu tehniku steganografije, dok je univerzalna upotrebljiva za bilo koji medij (svojstva su zajednička i usporediva).

Prvi alat statističke analize je *Chi-Square Attack* (χ^2) razvijen za detekciju tajnih informacija ugrađenih preko LSB alata [7]. Zasnovan je na principu postojanja *PoV* parova vrijednosti, tj. da L-bitni kanali boja mogu imati ukupno $P = 2^L$ različitih vrijednosti. Podjela na 2^{L-1} parova, koji se razlikuju u LSB-ovima, daje sve moguće uzorke susjednih bitova pojedinog LSB-a, a svaki ovaj par naziva se *PoV*. Npr. parovi 0 i 1 čine jedan *PoV* par jer njihova promjena LSB-a uzrokuje promjenu nule u jedinicu i jedinice u nulu. Isto vrijedi za ostale vrijednosti: 2 prelaze u 3, 3 u 2, 4 u 5, 5 u 4 itd. Tako npr. 8-bitnih kanali imaju ukupno 128 parova. Nadalje, distribucija parnih i neparnih vrijednosti ovih parova jednaka je kao 0/ 1 distribucija tajnih bitova ukoliko su iskorišteni svi mogući LSB-ovi [7]. Ideja χ^2 napada je usporedba navedene teoretski očekivane distribucije *PoV* parova sa distribucijom *PoV* parova originalnog objekta.

RS stegoanaliza (engl. *RS Analysis*) je metoda koja služi pri procjeni duljine informacija ugrađenih preko LSB tehnike [40]. Zasnovana je na činjenici da postoji veza između LSB ravnine i slike, tj. da je sadržaj svake bitne ravnine u vezi s ostalim ravninama. Prati kako se broj R (*Regular*) i S (*Singular*) grupa mijenja u ovisnošću s povećanjem duljine tajne poruke ugrađene unutar LSB ravnine. Odvija se podjelom unutar $G = (x_1, x_2 \dots, x_n)$ razdvojenih grupa po odabranih n susjednih piksela. Ova podjela vrši se uz pomoć funkcije

$$f(x_1, x_2 \dots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|, \tag{3-2}$$

gdje x_{i+1} i x_i predstavljaju vrijednosti susjednih piksela, a n broj ukupno odabranih [7, 40]. Što je veća vrijednost funkcije f, to je grupa G „glasnija“ [40]. Ova „glasnoća“ povećava se primjenom LSB tehnika, te ju je cilj promatrati i zabilježiti [40]. Kako bi se navedeno postiglo, definirane su funkcije koje se primjenjuju na pojedini piksel unutar određene vrijednosne domene. Npr. za 8-bitnu sliku piksel ima vrijednosti $P = \{0, 1, \dots, 255\}$. Uz primjenu *flipping* funkcije $F_1: 0 \Leftrightarrow 1, 2 \Leftrightarrow 3, \dots, 254 \Leftrightarrow 255$, *shifting* funkcije $F_{-1}: -1 \Leftrightarrow 0, 1 \Leftrightarrow 2, \dots, 255 \Leftrightarrow 256$, funkcije identiteta $F_0 = F_0(x), \forall x \in P$ te n-torke M s vrijednostima $\{-1,0,1\}$ (maska koja određuje koja funkcija će se primijeniti na koji piksel – -1 za F_{-1}, 0 za F_0 te 1 za F_1), G se klasificira unutar tri klase:

$$Regular\ G \in R_M \Leftrightarrow f\big(F_M(G)\big) > f(G), \tag{3-3}$$

$$Singular\ G \in S_M \Leftrightarrow f\big(F_M(G)\big) < f(G), \tag{3-4}$$

$$Unusable\ G \in U_M \Leftrightarrow f\big(F_M(G)\big) = f(G), \tag{3-5}$$

gdje R_M, S_M te U_M predstavljaju klase, a F_M i $f(G)$ ranije definirane funkcije [7, 40]. Na isti način klasificira se po grupama S_{-M}, R_{-M} te U_{-M}, ali uz primjenu maske $-M$ (komplement maske M) [7]. Tada vrijedi:

$$\frac{R_M + S_M}{T} \leq \frac{R_{-M} + S_{-M}}{T} \leq 1, \tag{3-6}$$

gdje T predstavlja ukupan broj G grupa [7]. Drugačije rečeno, slika se podijeli na n blokova iste veličine, R_M se definira kao udio blokova gdje se f povećava uz primjenu F_1, a S_M kao udio blokova gdje se f smanjuje. Iz ovoga slijedi $R_M + S_M \leq 1$ (analogno za R_{-M} i S_{-M} uz primjenu F_{-1}) [40]. Ukoliko slika ne sadržava tajne informacije, funkcije F_1 i F_{-1} na jednak način doprinose povećanju f vrijednosti G blokova, te slijedi

$$R_M \approx R_{-M} > S_M \approx S_{-M}, \tag{3-7}$$

a u suprotnom

$$R_{-M} - S_{-M} > R_M - S_M, \tag{3-8}$$

gdje su R_M, R_{-M}, S_M i S_{-M} gore definirane grupe [40]. Što je veća količina ugrađene poruke, manja je razlika između R_{-M} i S_{-M}, a veća između R_M i S_M [7, 40]. RS stegoanaliza pasivan je napad temeljen na detekciji prisustva tajne poruke.

3.2.3. Strukturna detekcija

Strukturna detekcija (engl. *structural detection*) se obavlja usporedbom metapodataka izvorne datoteke i stego-objekta [39]. Metapodatci su podaci o podacima koji služe kao dodatan opis datoteka, čime im se povećava vrijednost. Dijele se na opisne (npr. naslov, autor, opis), strukturne (npr. format, korištena kamera, izlaganje) i administrativne (npr. kada i kako se datoteka stvorila, tko smije pristupiti i sl.) [42]. Prilikom primjene određenih tehnika/ alata dolazi do zamjetnih promjena vrijednosti ovih atributa poput dubine boje, tipa i veličine datoteke, rezolucije, procesa kodiranja, brzine prijenosa (engl. *bit rate*) i dr. Don Caeiro i Sanjana S. proveli su napade na jpeg datoteke preko strukturne detekcije na alatima poput *Invisible Secrets, Our Secret, Quick Stego* i *Steg* [39]. Kao što je vidljivo iz [39] i tablice 3.1. , alat *Invisible Secrets* pokazao je zamjetnu

razliku unutar komentara i veličine datoteke. Osim toga, postojeći komentari su bili prepisani, a novi nadodani.

Tablica 3.1. Metapodatci originalne i stego datoteke, alat *Invisible Secrets*, prema [39]

ATRIBUT	**ORIGINALNA DATOTEKA**	**STEGO-DATOTEKA**
Datoteka	1440 x 900 JPEG (1.3 mega-piksela) 168854 B (165 KB)	1,440 × 900 JPEG (1.3 mega-piksela) 816342 B (797 KB)
JFIF verzija	1.01	1.01
Proces kodiranja	Baseline DCT, Huffman coding	Baseline DCT, Huffman coding
Bitova po uzorku	8	8
Komentari	CREATOR: gd-jpeg v1.0 (using IJG JPEG v62), quality = 90	%a2%8c0B!%a9%8e%ac%a2%cb%f5xN 1~%ae%fb%92%10]|%17%15%e3%fe%e 2%b9%14%a5%f2%e6%11%8f%a2%8b %b9%db%cch.%ff%0f%c4ew%18%8fkw %a7%8d%b0>v%cc%81%8b%c8Q%caf. %ba%d7A%9bR%ed%ca%b81

S druge strane, alat *Our Secret* pokazao je zamjetnu promjenu u veličini, te su, kao što je vidljivo iz tablice 3.2. , atributi poput rezolucije, procesa kodiranja, tipa datoteke i dr. ostali isti. Iako promjena veličine nije vjerodostojan pokazatelj prisustva steganografije, moguće je pretpostaviti da konačna razlika u veličini ovisi o veličini tajne informacije, te bi se, pri većim odstupanjima, moglo pretpostaviti da je nešto sumnjivo. Nadalje, alat *Quick Stego* pokazao je promjene na tipu datoteke, MIME (engl. *multi-purpose internet mail extension*) ekstenziji te veličini. Osim izmjene postojećih, pojedini atributi su nadodani ili uklonjeni (prema [39], pretpostavka je zbog promjene tipa datoteke). Neke od ovih promjena prikazane su u tablici 3.2.

Tablica 3.2. Metapodatci originalne i stego datoteke, alat *Our Secret*, prema [39]

ATRIBUT	**ORIGINALNA DATOTEKA**	**STEGO-DATOTEKA**
Datoteka	492 x 278 JPEG 22690 B (22 KB)	492 x 278 BMP 410382 B (401 KB)
JFIF verzija	1.02	-
Rezolucija	100 piksela/ None	-
Tip datoteke	JPEG	BMP
MIME tip	image/ jpeg	image/ bmp
DCT Encode verzija	100	-
Bitna dubina	-	24
Kompresija	-	None
Duljina slike	-	410328

Proces kodiranja	Baseline DCT, Huffman coding	-

Alat *Steg* je, s druge strane, napravio izmjenu u procesu kodiranja. Iz navedenog proizlazi da je strukturna detekcija pasivan napad, temeljen na otkrivanju prisustva steganografije.

3.2.4. Analiza stranih kanala

Otkrivanje sumnjivih nositelja i odabir ispravnog napada težak je proces. S obzirom na količinu i lakoću postavljanja podataka na Internetu, ponekad je puno bolje usredotočiti se na izvor. Stegoanaliza stranih kanala (engl. *side channel stegoanalysis*) odnosi se na otkrivanje steganografa na mrežnim stranicama putem analize ponašanja [6]. Steganograf se lako može razotkriti putem nepažljivog ponašanja, koje analiza stranih kanala proučava. Uz činjenicu da se većina steganografskih tehnika temelji na algoritmima i ignorira proces selekcije lažnih objekata, steganograf može postaviti puno nasumičnih slika koje je teško analizirati, što po sebi privlači pozornost i sumnju. Osim toga, svaki korisnik društvenih mreža ima svoju povijest, koja može poslužiti kao dobar indikator (ne)sumnjivog ponašanja. Obično, ovaj povijesni slijed predstavlja karakteristike korisnika poput interesa, životnog stanja, osjećaja, događanja i sl. [6]. Običan korisnik imati će objave više manje vezane jednu uz drugu (konzistentne, „prirodne“ slike), dok će se nepažljivi steganograf više temeljiti na tehnici steganografije, tj. neće biti konzistentan u objavama (ili će odstupati u određenoj mjeri).

3.2.5. Metode obasipanja

Saurabh Choudhury, Amritha P.P i M. Sethmadhavan predložili su metode obasipanja (engl. *showering mechanism/ showering methods*) za uništavanje zloćudnih kodova unutar slika [43]. U sklopu ovog mehanizma uveli su dva napada, zamjenu LSB bitova (engl. *LSB Bit Flipping*) i dvostruku ugradnju (engl. *Double Stegging*). Nadalje, u svrhu analize i procjene rezultata uključili su analizu histograma (engl. *histogram analysis*), alat *StegExpose* te RS stegoanalizu. Zamjena LSB bitova jednostavna je metoda kod koje se mijenjaju svi LSB-ovi (prelaze iz nule u jedinicu i obrnuto), što osigurava uništavanje bilo kojih informacija temeljenih na LSB ugradnji. Dvostruka ugradnja je postupak koji se temelji na dodavanju nasumičnih podataka na već postojeće, a svrha je stvaranje nečitkosti na izvorno kodiranu poruku. *StegExpose* je analitički alat zasnovan na detekciji LSB steganografije unutar digitalnih slika. Kao što je vidljivo iz [43], iako RS analiza nije uspjela detektirati prisustvo steganografije, nakon primjene metoda obasipanja (slika 3.4.), skrivene programe nije bilo moguće iščitati.

Postotak iskorištenosti slike [%]	RS stegoanaliza [%]
100	66.32
50	32.45
30	26.23
5	17.18

Postotak iskorištenosti slike [%]	Metoda obasipanja	RS stegoanaliza [%]
100	Zamjena LSB bitova	<1
100	Dvostruka ugradnja	<5
50	Zamjena LSB bitova	<1
50	Dvostruka ugradnja	<5
30	Zamjena LSB bitova	<1
30	Dvostruka ugradnja	<3
5	Zamjena LSB bitova	<1
5	Dvostruka ugradnja	<3

Slika 3.4. Usporedba provedbe *RS Stegoanalysis* metode nad običnim i *showered* stego-objektom, prema [43]
Rezultat RS analize predstavlja vjerojatnost prisustva steganografije.

3.2.6. Analiza potpisa

Analiza potpisa (engl. *signature steganalysis*) prati promjenu svojstava stego-objekta ovisno o primjeni pojedinih tehnika i/ ili alata [7]. Većina alata stvara određenu degradaciju ili ponovljene uzorke koji predstavljaju „potpis“ korištenog algoritma. Steganografski alat *Hide & Seek* stvara stego-objekt koji sadrži vrijednosti piksela djeljive s 4, a alat *Jpegx* ugrađuje tajnu poruku na kraj jpeg datoteke, koja prethodi s hex kodom „5B 3B 31 53 00“ [7]. Cilj je otkrivanje je li objekt bio izmijenjen (sadrži li sumnjive uzorke), na temelju čega se odlučuje je li potrebna daljnja analiza i u kojoj mjeri. Pasivan je napad temeljen na prijašnje definiranim uzorcima. Kako su univerzalni napadi većinom teški za provesti, analiza potpisa predstavlja jednostavnije rješenje detekcije, ukoliko se uzme za pretpostavku da su poznati potpisi promatranih algoritama ugradnje.

4. PROGRAMSKO RJEŠENJE ZA PROVOĐENJE NAPADA NA STEGANOGRAFSKE SUSTAVE

Većina steganografskih napada vezana je uz konkretne algoritme i tehnike, tj. alate. Ne postoji univerzalan napad koji će probiti svaku tehniku, niti metode koje će pružiti stopostotno točne rezultati (engl. *false positives*, *false negatives*). Mogu se promatrati potpisi, uzorci i izobličenja, ali u tom trenutku je već kasno jer je alat vrlo vjerojatno postojao dugo vremena i već izvršio svoju zadaću. S druge strane, stavljanje pažnje na pojedinu tehniku može prikazati, a k tome i pružiti, mogućnost sistematske analize. Osim toga, ova sistematska analiza omogućuje klasifikaciju alata prema korištenim tehnikama, a time i sužava napade na one praktične. Moguće je dakako odlučiti se za alat koji kombinira sve poznate steganografske alate, no vjerojatnost da korisnik nije napravio bar neke promjene je malena – proširenja u načinu ugradnje, kriptiranje poruke i sl. Osim toga, napadi trebaju provjeriti i otpornost korištene tehnike, kao i njenu učinkovitost.

U nastavku je opisana odabrana implementacija, testiranje te osvrt na moguća poboljšanja tj. nedostatke.

4.1. Opis aplikacije i specifikacija zahtjeva

Kako je napada mnogo te implementacija praktično korisnog rješenja vrlo vremenski zahtjevna i teška, aplikacija stavlja fokus na demonstrativni „omotač“ oko odabranih steganografskih tehnika, napada na te tehnike, te mogućnost ugradnji pomoću nekih od njih. Praktično je bolje imati više napada, kao što su npr. vizualni, analitički, napadi uništenja i dr., te omogućiti njihovo istovremeno izvršavanje, što se može postići višestrukom obradom podataka (engl. *multiprocessing*). Ova višestruka obrada zasnovana je na sposobnosti raspoređivanja poslova među procesorima čime se ostvaruje brže vrijeme izvođenja, uzimajući dakako u obzir da okolina na kojoj se izvodi aplikacija to podržava. Osim toga, aplikacija bi trebala omogućiti lako dodavanje novih alata i modula. Pri tome bi se alati trebali moći provesti i „izvan“ aplikacije, kao zasebne jedinice, bez npr. grafičkog sučelja, što bi omogućilo neovisnost uporabe o aplikaciji. Zahtjevi na provođenje pojedinih napada nisu veliki, a primarno je potrebna snaga procesora radi manipulacije datoteka. Osim toga, treba voditi brigu o brzini i efikasnosti korištenih algoritama (npr. sporo izvođenje na slikama velikih formata). Nadalje, zbog dominantnosti slika kao medija za prijenos podataka, aplikacija se temelji na slikovnim formatima bmp, jbg te png. U nastavku slijede korištene tehnologije i biblioteke.

4.1.1. Python i KivyMD

Kao najbolje rješenje za opisano pokazao se programski jezik Python. Pruža mnoštvo biblioteka za obradu slika, rada s operativnim sustavima, datotekama te raznolikim vrstama podataka. Python je objektno-orijentirati programski jezik visoke razine, čija priroda nudi lako dodavanje novih modula i funkcija, a k tome je i temeljito dokumentiran. Za razvojno programsko okruženje odabran je *PyCharm* radi opsežnih mogućnosti, a za izradu korisničkog sučelja međuplatformski okvir *KivyMD*. *KivyMD* predstavlja skup *Material Design* grafičkih programa (engl. *widget*). Temeljen je na *Kivy* okviru, a za razliku od njega predstavlja atraktivnije sučelje te nudi više opcija za izmjenu grafičkih objekata.

4.1.1. Pillow (PIL) i Matplotlib

Pillow (*PIL*) je biblioteka koja omogućuje obradu datoteka. Korisna je za arhiviranje slika, izradu minijatura (engl. *thumbnails*), mijenjanje formata datoteke, ispis slika i dr. [44] Pruža razne funkcije poput filtiranja, rotacije, konvolucije, izvlačenja statistika i dr. operacija. Unutar aplikacije je korištena pri čitanju i spremanju slika s odabranim tipom formatiranja te izmjeni piksela. *Matplotlib* je biblioteka za stvaranje statičnih, animiranih i interaktivnih vizualizacija [45]. Omogućuje stvaranje grafikona, prilagođavanje vizualnog izgleda, spremanje i dr. funkcija.

4.1.1. Multiprocessing, OpenCV (cv2) i NumPy

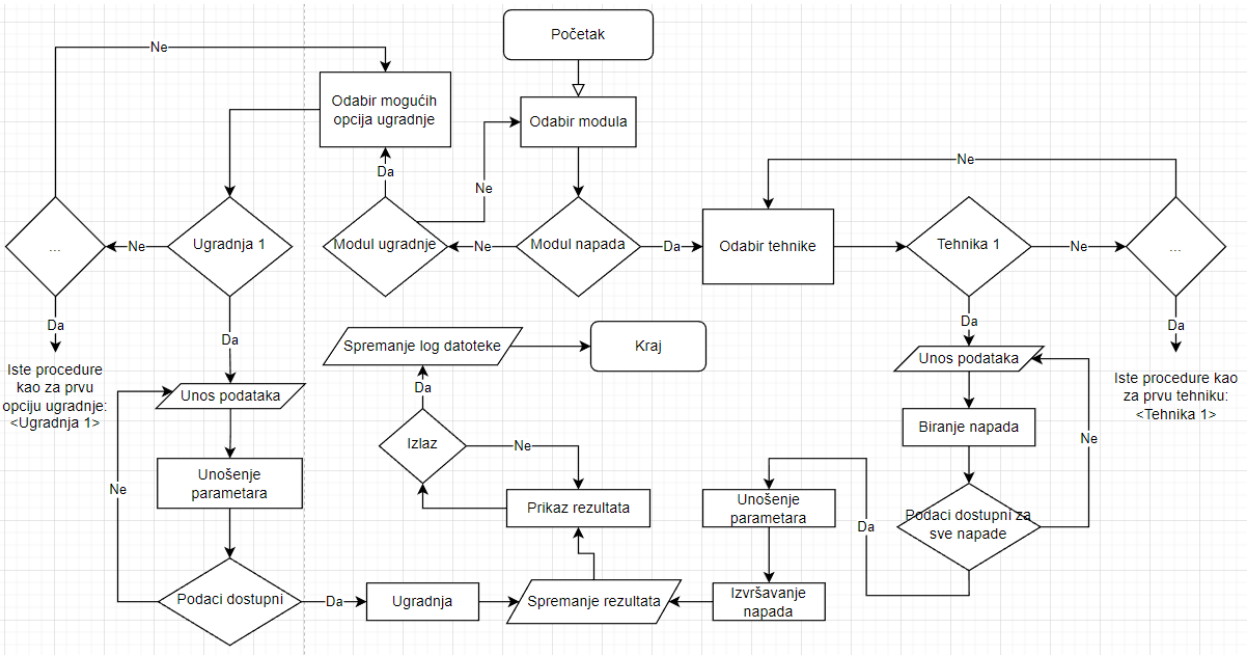
Multiprocessing je biblioteka koja omogućuje raspoređivanje poslova na dostupne procesorske jedinice, tj. njihovo iskorištavanje. Pruža komunikaciju, sinkronizaciju, međusobno dijeljenje stanja, memorije i dr. resursa [46] *OpenCV* (*cv2*) je biblioteka otvorenog koda. Primarno se primjenjuje za potrebe računalnog vida (engl. *computer vision*), strojnog učenja (engl. *machine learning*) i obrade slika (engl. *image processing*) [47]. Podržava jezike poput Jave, C++-a i Python-a. *NumPy* je C++ biblioteka visoko optimizirana za provođene raznih numeričkih operacija, primjenjiva primarno unutar podatkovnih znanosti (engl. *data science*) [48].

4.1.2. Pyexiv2 (exiv2) i Os

Pyexiv2 (*exiv2*) je višeplatformska C++ biblioteka te program za upravljanje metapodacima slika [49]. Omogućuje brz i jednostavan pristup za čitanje raznih tipova metapodataka. Koristi se unutar mnogih projekata poput GIMP, darktable, showell, GwenView i Luminance HDR. Podržava Exiv, IPTC i XMP tipove metapodataka. *Os* je biblioteka koja pruža funkcionalnosti ovisne o operativnom sustavu, kao što su otvaranje i manipuliranje datotekama i direktorijima [50].

4.2. Dijagram toka

Dijagram toka aplikacije prikazan je na slici 4.1.

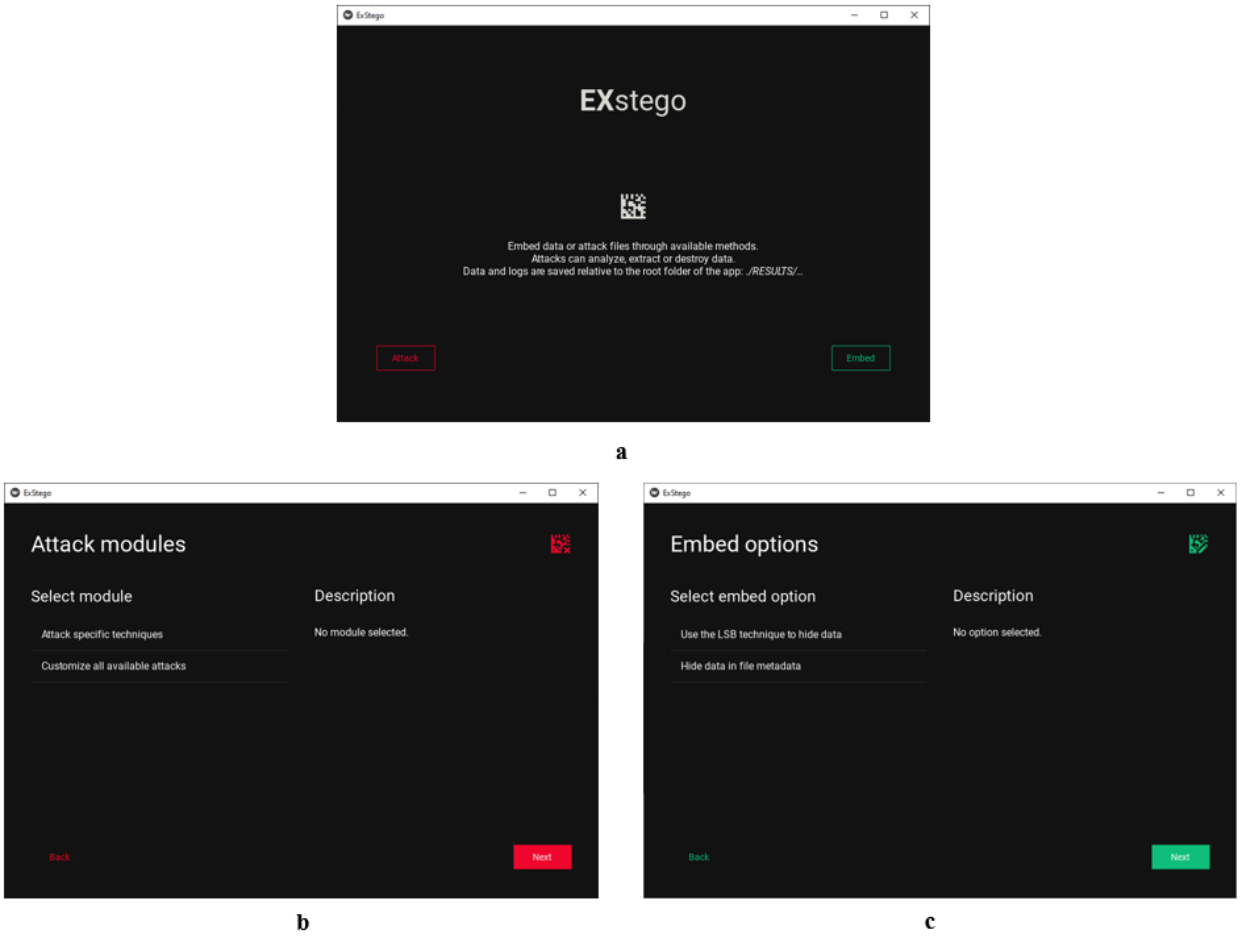


Slika 4.1. Dijagram toka aplikacije

Aplikacija započinje odabirom modula (modul ugradnje ili modul napada). Svaki modul nudi tehnike (npr. ugradnja 1 ili tehnika 1) koje imaju isti tok rada, gdje se kod ugradnje biraju dostupni podaci i parametri, a kod napada na tehnike svi dostupni napadi, a naknadno parametri za svaki od njih. Nadalje, prije izvođenja, provjera se dostupnost svih podataka. Za ugradnju je potrebna jedna datoteka, za uništenje stego-objekt, a za usporedbu su potrebni i original i stego-objekt. Nakon izvršavanja svih poslova, rezultati se prikazuju unutar povratnog prozora te dodatno spremaju unutar jedinstveno stvorenog direktorija. Nakon izlaska iz aplikacije sprema se i log datoteka koja sadrži detaljnije informacije o korištenim alatima, što pruža uvid u moguće greške tokom izvođenja.

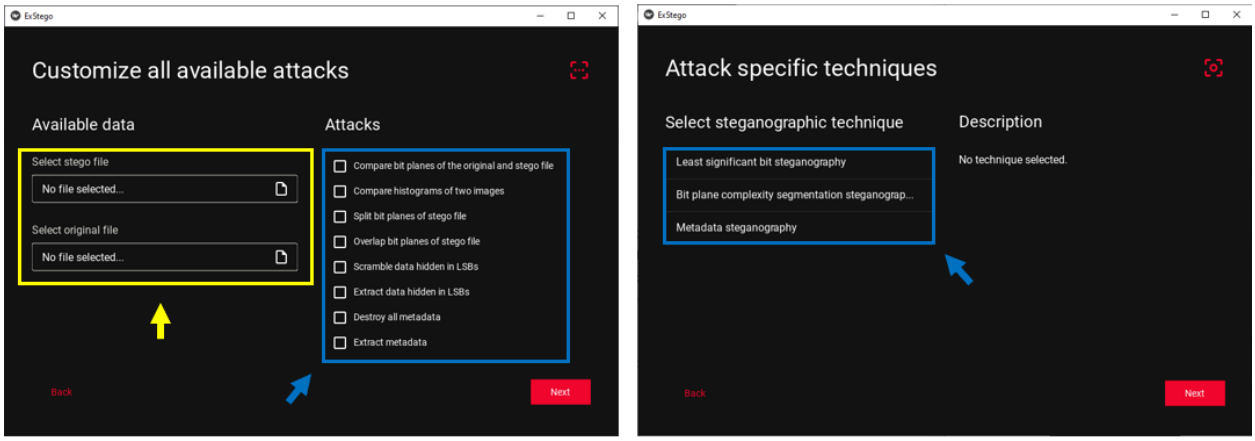
4.3. Mogućnosti i rad aplikacije

Mogućnosti i rad aplikacije vezan je uz odabrani modul, čiji se odabir (slika 4.1.) nudi na početnom prozoru (sličica a), a oni su modul napada prikazan na sličici b te modul ugradnje prikazan na sličici c.



Slika 4.1. Glavni moduli aplikacije

Modul napada dijeli se na dva podmodula (slika 4.2.). Podmodul *Customize all available attacks* (sličica a) omogućuje odabir svih dostupnih napada, dok podmodul *Attack specific technique* (sličica b) filtrira napade ovisno o tome za koju tehniku su primarno namijenjeni. Usporedbom ovih modula s dijagramom toka, *Customize all available attacks* predstavlja specijaliziranu tehniku koja obuhvaća sve napade, s time da je procedura provođenja ista, uz puno veći odabir. Navedena implementacija pruža fleksibilnost i sistematičnost bez obzira koja je tehnika (ili varijacija tehnike) korištena. Ovo omogućuje izvršavanje napada sukladno potrebama korisnika, npr. je li poznat stego-objekt, tehnika i sl. (napad je specifičan), ili se pak žele odabrati svi napadi i time osigurati uništenje poruke ili otkriti postojanost sumnje (nije poznato ništa o načinu ugradnje, korisnik svakako mora provesti napade na sve tehnike). Nadalje, prije nego li je moguće odabrati napad, provjerava se postoje li podaci nužni za izvršavanje, čiji se odabir nalazi unutar žutog kvadrata na sličici a. Npr. za analitičke napade potrebna su i originalna i stego datoteka, dok je za ostale jedino potreban stego-objekt. Ukoliko nužne datoteke nisu priložene, odabir neće biti moguć.



a b

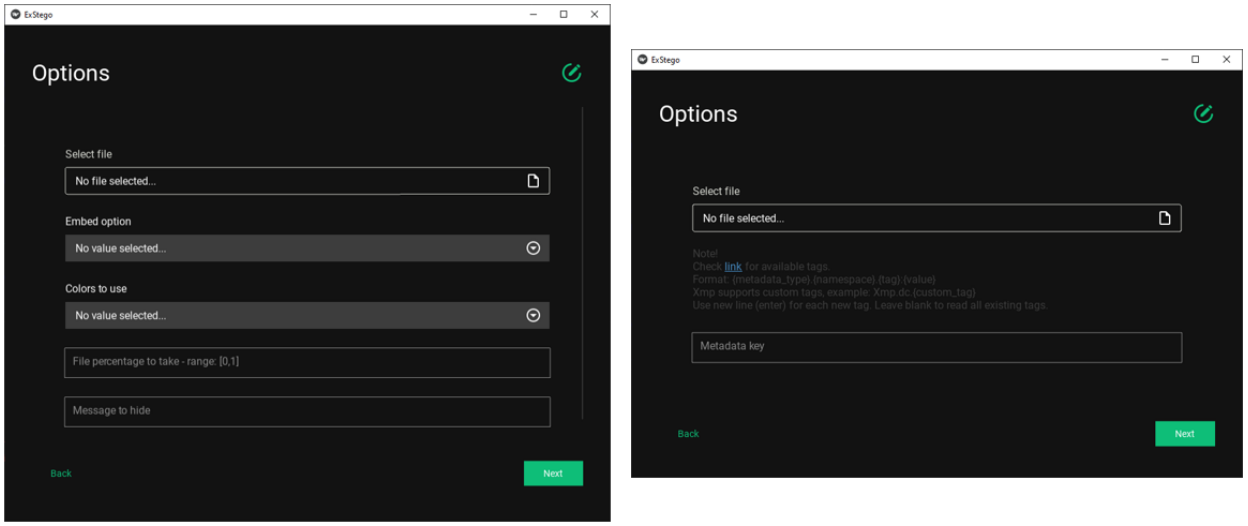
Slika 4.2. Podmoduli napada

Svi dostupni napadi označeni su unutar plavog kvadrata na sličici a, a oni su: usporedi bitne razine originalne datoteke i stego objekta (*Compare bit planes of original and stego file*), usporedi histograme (*Compare histograms*), razdvoji bitne razine stego-objekta (*Split bit planes of stego file*), preklopi bitne razine stego-objekta (*Overlap bit planes of stego file*), uništi podatke skrivene unutar najmanje značajnih bitova (*Scramble data hidden in LSBs*), iščitaj podatke skrivene unutar najmanje značajnim bitovima (*Extract data hidden in LSBs*), uništi sve metapodatke (*Destroy all metadata*) te iščitaj metapodatke (*Extract metadata)*. Temeljeni su na klasifikaciji napada unutar poglavlja 3.1, čime objedinjuju npr. statističke napade, napade uništenja, napade poznate tehnike, napade preko stego-objekta i dr. Tehnike koje integriraju ove napade prikazane su unutar plavog kvadrata na sličici b, a to su LSB tehnika (*Least significant bit steganography*), BPCS tehnika (*Bit plane complexity segmentation steganography*) te tehnika metapodataka (*Metadata steganography*). Tablica 4.1. prikazuje integraciju dostupnih napada unutar namijenjenih tehnika.

Tablica 4.1. Integracija dostupnih napada unutar namijenjenih tehnika

BPCS tehnika	**LSB tehnika**	**Tehnika metapodataka**
Usporedi bitne razine originalne datoteke i stego-objekta	Uništi podatke skrivene unutar najmanje značajnih bitova	Uništi sve metapodatke
Razdvoji bitne razine stego-objekta	Iščitaj podatke skrivene unutar najmanje značajnim bitovima	Iščitaj metapodatke
Preklopi bitne razine stego-objekta	Usporedi histograme	-

S druge strane, modul ugradnje omogućuje ugradnju uz pomoć dvije tehnike, LSB tehnike i tehnike metapodataka. Odabirom tehnike odlazi se na prozor koji traži datoteku unutar koje se tajna poruka želi sakriti te nudi odabir parametara (slika 4.3.), koji određuju konačan tok/ ključ ugradnje. Sličica a prikazuje odabir parametara LSB tehnike, a sličica b tehnike metapodataka.



a **b**

Slika 4.3. Prozori ugradnje; LSB tehnika (a), tehnika metapodataka (b)

Parametri su vezani uz specifične tehnike ugradnje. U sklopu ovoga ugradnja unutar LSB-a te napadi njihova uništenja i iščitavanja omogućuju odabir RGB komponenata (crvena, zelena, plava), postotka te tehnike kodiranja (napad usporedbe histograma ne nudi dodatne parametre). RGB komponente određuju boje kroz koje će algoritam prolaziti prilikom provođenja napada ili ugradnje. Npr. ugradnja odabirom para (plava, crvena) ugrađuje željenu poruku redoslijedom plava pa crvena. Prilikom provođenja napada iščitavanja poruke odabirom para (crvena, plava) neće se dobiti dobar rezultat jer će se bitovi poruke čitati krivim redoslijedom. Drugim riječima, njihov odabir definira tok na razini boja. S druge strane, postotak i tehnika kodiranja definiraju tok na razini piksela. Postoje dvije tehnike kodiranja, a to su sekvencijalna opcija (*Sequential*) te postepena opcija (*Scatter option*). Indeks piksela prilikom čitanja i ugradnje odabire se na sljedeći način:

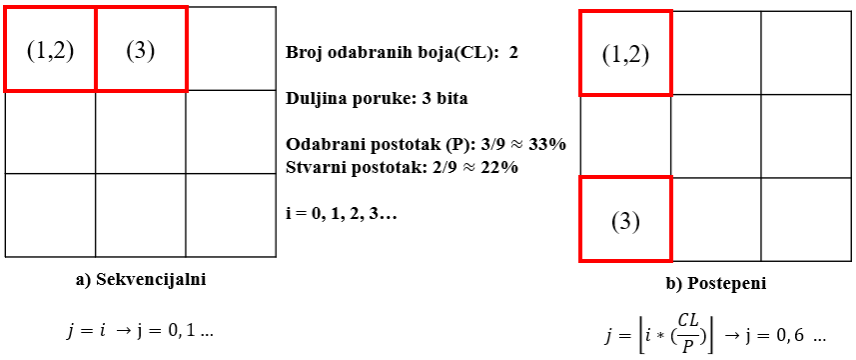
$$i \in [0, 1, 2, \dots IL - 1], \tag{4-1}$$

$$\text{Sekvencijalni: } j = i, \tag{4-2}$$

$$\text{Postepeni: } j = \left\lfloor i * \left(\frac{CL}{P}\right) \right\rfloor, \tag{4-3}$$

gdje i predstavlja jedan od indeksa piksela slike, IL ukupan broj piksela slike, j novi indeks, CL broj odabranih RGB komponenata, a P odabrani postotak u decimalnom obliku. Sekvencijalan način prolazi kroz piksele korakom od 1 piksela, dok postepeni način podešava korak na temelju postotka te odabranog broja RGB komponenti. Npr. za sliku od ukupno 9 piksela. Prilikom provođenja napada čitanja s odabranim postotkom od 33%, jedne RGB komponente (recimo

crvene) te sekvencijalnog načina rada, algoritam će pročitati vrijednost crvene komponente od 33% piksela, tj. 3 piksela, što će biti i ukupna duljina poruke (3 LSB-a, 3 bita). S druge strane (slika 4.4.), odabirom npr. dvije RGB komponente te sekvencijalnog načina rada (sličica a), algoritam neće pročitati 33% svih piksela, jer će se prilikom čitanja morati zadržati do 2 puta na svakom pikselu. Ukupna duljina poruke iznositi će i dalje 3 bita (33% ukupnog broja piksela), ali će biti obuhvaćena unutar 2 piksela (22% ukupne slike), umjesto 3. Postepeni način radi na istom principu (sličica b), s time da je i korak podešen s obzirom na odabrane parametre, a u ovom slučaju iznositi će 6 piksela. Algoritam je i kao i kod sekvencijalnog načina prošao kroz 2 piksela te je duljina poruke i dalje 3 bita.



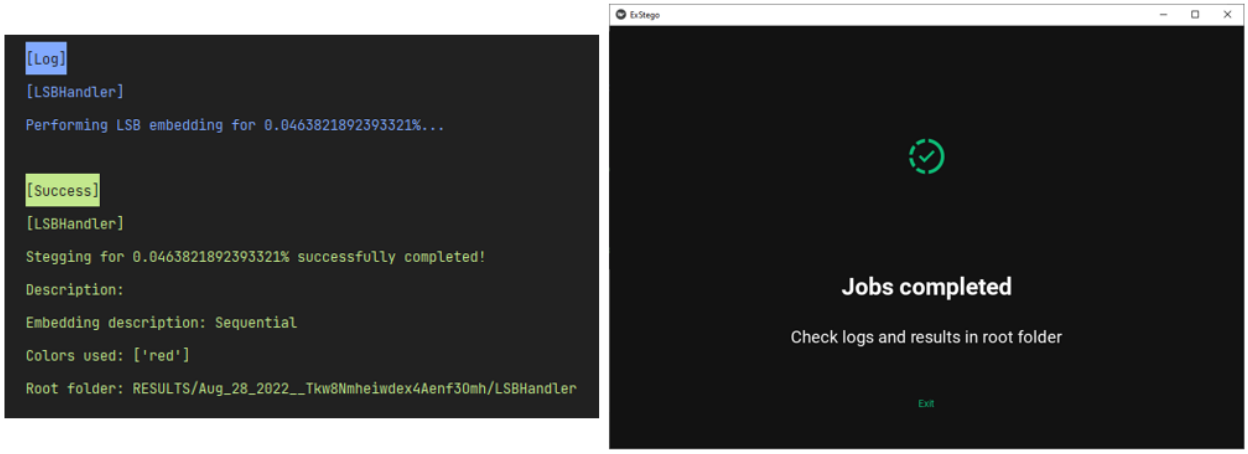
Slika 4.4. Rad LSB algoritma ugradnje i čitanja

S druge strane, napad uništenja obuhvatiti će odabrani postotak kao postotak piksela, koji god parametri odabrani. Navedeno se korigira povećanjem iteracija, npr. za uvjete jednake kao i na prethodnom primjeru. Ukoliko su odabrane dvije RGB komponente, broj iteracija iznositi će ukupno 6 (umjesto 3). Unutar ovih šest iteracija, uništenje se zadržava 2 puta na svakom pikselu, te će se obuhvatiti ukupan početni postotak od 33%. Ovim je omogućeno da napad uništenja nije isključivo vezan za dostupne ključeve ugradnje, već na željenu razinu uništenja, što pruža veću fleksibilnost. Drugim riječima, odabirom svih RGB komponenata i postotkom od 100%, svi podaci bi trebali biti uništeni, neovisno o LSB alatu, bio on dostupan unutar aplikacije ili ne.

Napadi zasnovani na BPCS tehnici omogućuju odabir bitnih razina (*Bit planes to use*) te RGB komponenata (*Color channels to use*). Napadi usporedbe bitnih razina originalne datoteke i stego-objekta i podjele bitnih razina stego-objekta nude odabir oba parametra, dok napad preklapanja bitnih razina stego-objekta jedino odabir bitnih razina jer algoritam koristi sve RGB komponente. Cilj je izvući informacije (ili dokazati njihovo postojanje) poput skrivenih slika koje se mogu nalazi unutar pojedinih (ili svih) bitnih razina.

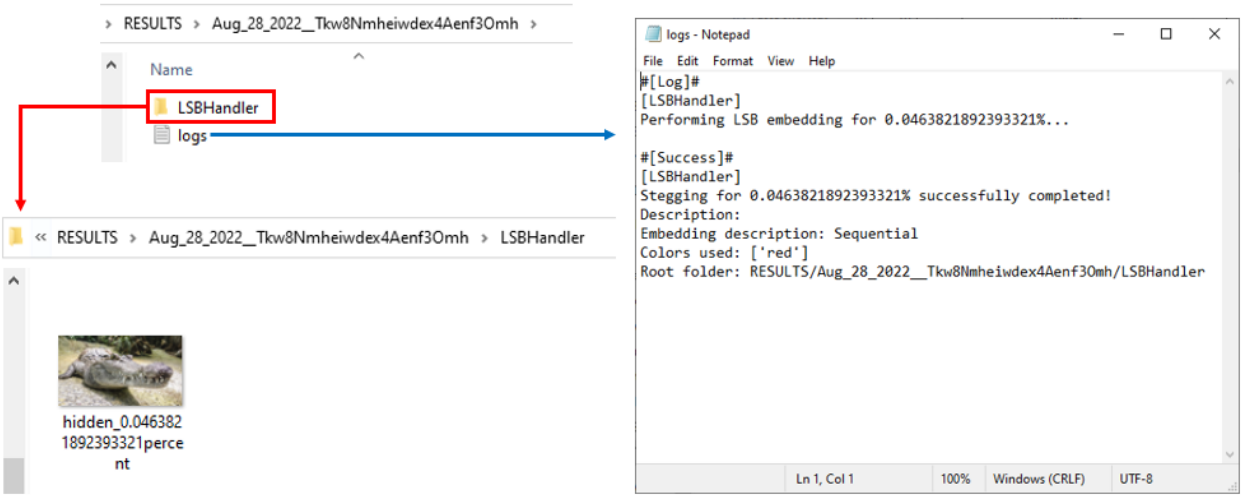
Dostupni parametri za tehniku metapodataka su ključevi koji se žele iščitati tj. unutar kojih se želi sakriti poruka. Uništenje upisuje nasumične znakove na sve dostupne metapodatke. Obuhvaćeni su Exif, IPTC te XMP tipovi metapodataka.

Sve povratne informacije vezane uz korištene alata nalaze se, osim na prozoru rezultata aplikacije i unutar direktorija jedinstvenog naziva. Osim toga, ukoliko je potrebno, moguće je unutar koda uključiti bilježenje log poruka tokom izvođenja što nudi uvid u moguće greške. Npr. za ugradnju korištenjem LSB tehnike, povratne informacije unutar aplikacije prikazane su na slici 4.5. , gdje sličica a predstavlja log poruke tokom izvođenja, a sličica b konačni prozor.



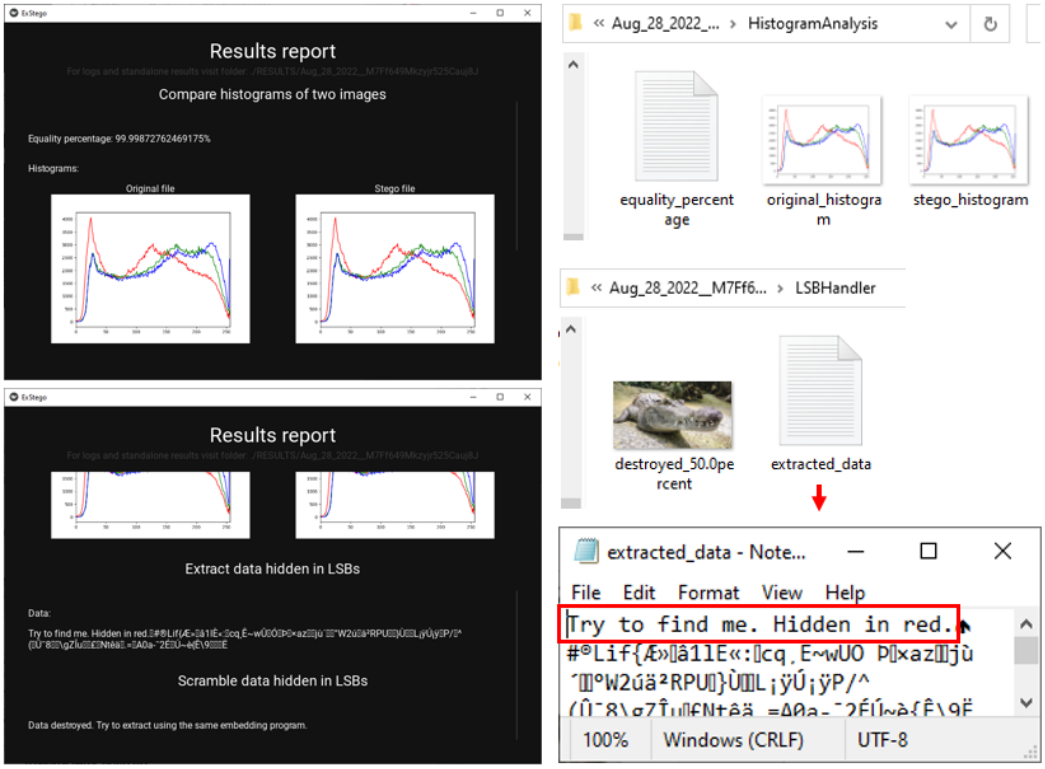
Slika 4.5. Povratne informacije tokom izvođenja aplikacije

S druge strane, slika 4.6. prikazuje povratne informacije izvan aplikacije.



Slika 4.6. Povratne informacije izvan aplikacije

Nakon što postoji slika sa skrivenim podacima, mogu se odabrati napadi koji se žele provesti. Npr. u ovom slučaju za postotak dekodiranja može se uzeti sve od 0.05% pa do 100%, jer je prijašnje bila poznata činjenica da je tajna poruka zauzela oko 0.0464%, stoga je unutar tih postotaka sigurno da će poruka biti potpuno čitljiva. S druge strane, ukoliko bi se odabrao postepeni način ugradnje, postotak bi morao biti jednak postotku čitanja, jer bi indeksi piksela u suprotnom bili netočni (različit ključ/ tok). Rezultati i rad napada ove tehnike prikazani su na slici 4.7. Npr. datoteka naziva „destroyed_50percent“ predstavlja novu 50% uništenu sliku, a izdvojeni podaci „extracted_data“ odnose se na original. Budući da se za napad uništenja odabrala ista metoda i RGB komponente kao i kod ugradnje, poruku neće biti moguće dekodirati na novoj slici.



Rezultati unutar aplikacije **Rezultati izvan aplikacije**

Slika 4.7. Rezultati napada na LSB tehniku

Napadi i ugradnje ostalih tehnika odvijaju se na isti način, dajući različite rezultate ovisno o odabranim metodama i parametrima.

4.3.1. Korištenje i testiranje aplikacije

Testiranje je prvotno provedeno na tehnike omogućene iz aplikacije. Tablica 4.2. predstavlja postotak sličnosti histograma originala i dobivene stego slike pri različitim postotcima ugradnje, a boje označavaju unutar kojih RGB komponenti se poruka ugradila. Usporedba histograma

izračunata je uz pomoć cv2 biblioteke preko korelacijske metode (ukupno postoje 4, oznaka $d(H_1, H_2)$), čija se vrijednost računa relacijama

$$d(H_1, H_2) = \frac{\Sigma_1 (H_1(I) - \bar{H}_1)\,(H_2(I) - \bar{H}_2)}{\sqrt{\Sigma_I (H_1(I) - \bar{H}_1)^2\ \Sigma_I (H_2(I) - \bar{H}_2)^2}}, \tag{4-4}$$

$$\bar{H}_k = \frac{1}{N} \sum_J H_k(J)\,, \tag{4-5}$$

gdje H_1 i H_2 predstavljaju histograme koji se uspoređuju, I i J indekse vrijednosti, a N broj „spremnika“ (engl. *bins*) [51]. Prilikom ugradnje korištena je LSB tehnika. P označava postepenu metodu, a S sekvencijalnu. NAN označava grešku (engl. *not a number*).

Tablica 4.2. Usporedba histograma pri različitim parametrima, LSB tehnika

Postotak ugradnje [%]	**Postotak sličnosti, histogram [%]**					
	Crvena		**Crvena, zelena**		**Crvena, zelena, plava**	
	S	**P**	**S**	**P**	**S**	**p**
5	99.4162	99.8920	90.5908	99.94437	99.73669	99.9549
10	98.4876	99.7521	99.153	99.94	99.37455	99.9398
20	94.1798	99.3342	97.8112	100.1274	98.65089	100.0460
40	84.1825	97.3915	91.4081	101.9727	95.6921	100.9754
50	79.10765	95.3394	86.7245	104.3891	93.3994	102.0367
75	67.682	79.5147	78.83339	131.669	85.5893	108.8687
100	25.3092	25.4918	69.25429	NAN	81.0466	144.3073

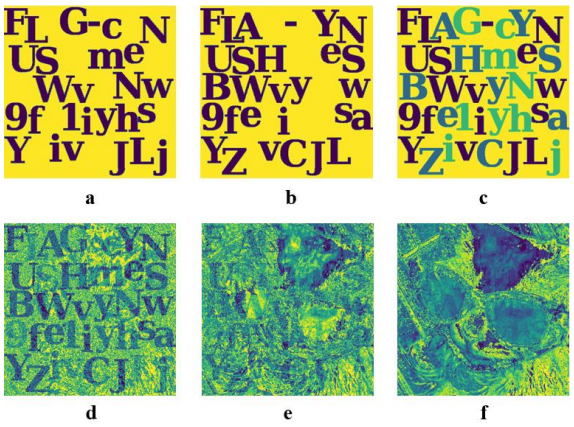
Kao što se može vidjeti iz tablice, povoljnije je uzeti sve RGB komponente za ugradnju poruke. Postepena opcija daje slične rezultate, iako korištenjem više kombinacija RGB komponenti napad prestaje biti vjerodostojan. Za odabir jedne boje, postotci su uvećani za najviše oko 16. Testiranje uz uništenje poruke bilo je uspješno za sve provedene metode, čak i ako nisu odabrani isti parametri. Povezujući navedeno s činjenicom da funkcija dekodiranja poruke skupove bitova povezuje po grupama od 8 (radi pretvaranja u znakove), lako je pretpostaviti da će se pojedini znakovi poruke drastično izmijeniti.

Drugi dio testiranja odnosio se na provođenje napada na BPCS tehnike. Kao što je vidljivo iz specifičnog problema [52], skrivanje unutar bitnih razina ovisi (unutar jednih tipova implementacija) o odabranoj razini te vezama između ostalih komponenti. Razlaganjem slike primijećeno je kako se pojavljuju određeni uzorci unutar 1. bitne razine. Navedeno se može vidjeti

na slici 4.8. , gdje crveno označene zone upućuju na prisustvo tajne poruke. Ostale komponente (plava i zelena) vizualno nisu upućivale na prisustvo informacija.

Slika 4.8. Usporedba izvorne slike i njene 1. bitne razine unutar crvene komponente

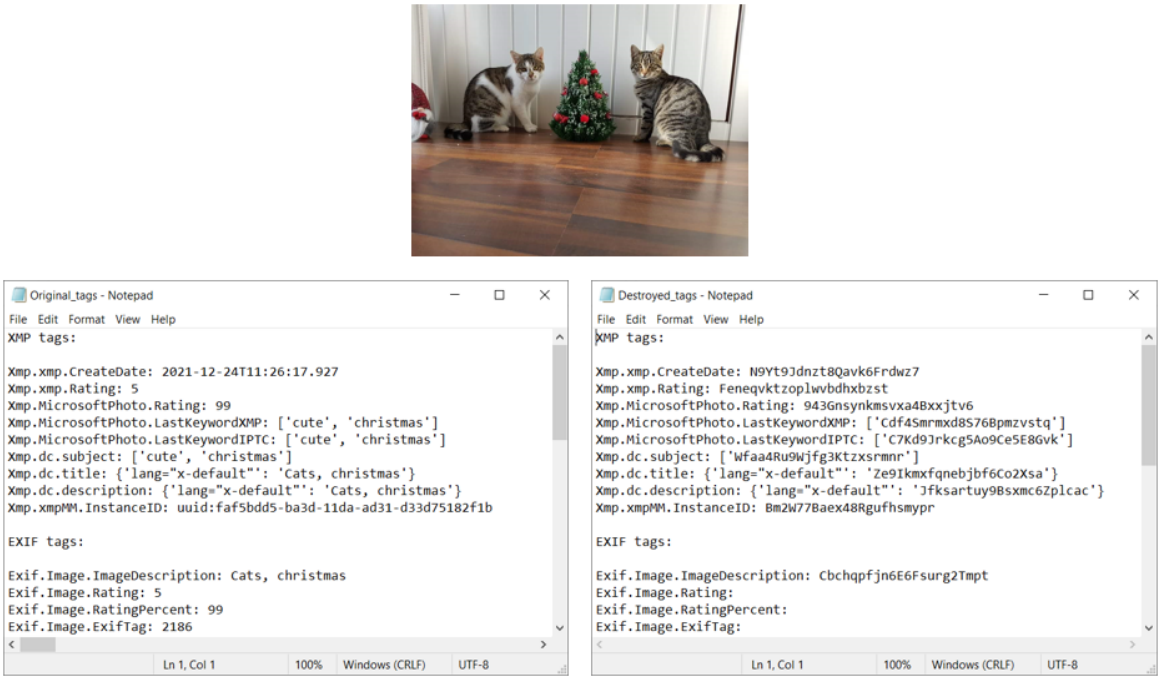
Daljnjim razmatranjima, promatrana su preklapanja bitnih razina (slika 4.9.). Iako je jedino crvena komponenta upućivala na prisustvo tajne poruke, bilo bi neispravno isključiti da se nešto krije i u ostalim komponentama. Osim toga, implementirana metoda preklapanja ograničena je na iskorištavanje svih RGB komponenata, dok je kod analize odabir proizvoljan. Testiranjem su promatrana sva moguća preklapanja (sve n-torke niza [1,…,8]), te kao što se može zaključiti iz slike, potrebno je „pogoditi“ koje od razina uistinu sadržavaju poruku (tj. njene dijelove). Ukoliko se preklapanje provelo za nedovoljan broj razina (recimo ako je promatrana 1. ili 2. bitna razina, sličice a i b), poruka neće biti potpuna. S druge strane, ukoliko se preklapanje provelo za razine koje ne sadržavaju nikakve informacije, dolazi do šuma (sličice d, e i f; preklapanje prvih 4, 5 i 6 bitnih razina). Utvrđeno je da se poruka nalazi unutar prve dvije bitne razine (sličica c, njihovo preklapanje), te je uspješno iščitana. Pokušaj uništenja poruke preko LSB napada pokazao se neuspješnim, što pokazuje otpornost na izmjenu LSB bitova.



Slika 4.9. Preklapanje bitnih razina, prva (lijevo), druga (sredina), prve dvije (desno)

Čitanje i uništenje metapodataka je također bilo uspješno (slika 4.10.). Metapodatke je bilo moguće i izmijeniti, no treba pripaziti na tip podatka koji se pohranjuje. Kako aplikacija jedino

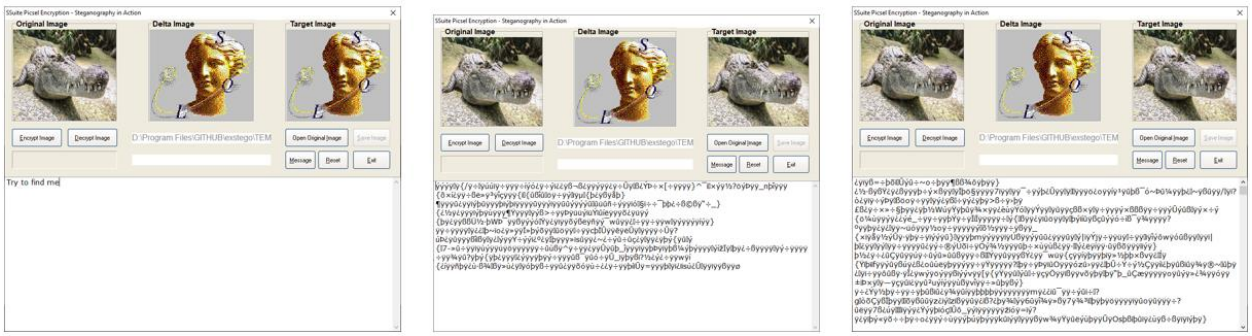
podržava ugradnju riječi (engl. *string*), ključ unutar kojeg se ugrađuje mora držati isti tip. Isto vrijedi i za uništenje te su stoga, kao što se može vidjeti na sličici b, određeni metapodaci prazni. Osim toga, nedostatak je što su pojedini ključevi lako vidljivi i unutar drugih lakše dostupnih alata, kao što je npr. prozor detalja pri pregledu svojstava unutar operacijskog sustava Windows.



a) Originalni metapodaci **b) Uništeni metapodaci**

Slika 4.10. Čitanje i uništavanje metapodataka

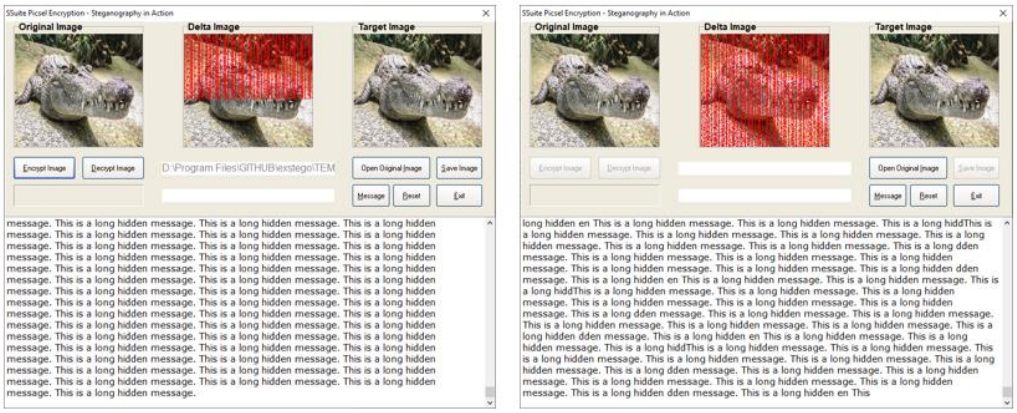
Nadalje, provedeni su napadi na poruke i medije ugrađene alatom *SSuite Picsel* [53] (slika 4.11. Alat koristi slike formata bmp prilikom ugradnje, te radi na principu da nakon kodiranja stvara ključ-sliku naizgled originalnoj. Ovime se postiže da je čitanje poruke moguće isključivo uz prisustvo originala, ključ-slike i alata. Razmatrana je mogućnost uništenja poruke pri provođenju LSB napada na ključ-slici, uz različite veličine tajne poruke. Kao što je vidljivo iz sličica a, b i c, iščitavanje je moguće jedino uz izvorni ključ, neovisno o duljini poruke (bilo kakve izmjene doprinose prevelikom šumu). Uništenje je bilo uspješno za 1%, uz bilo koje komponente boja i metode (sekvencijalna ili postepena). Zaključeno je kako je metoda izrazito osjetljiva na bilo kakve direktne promjene ključa.



a) Skrivena poruka **b) Iščitavanje iz 1% uništenog ključa** **c) Iščitavanje iz 50% uništenog ključa**

Slika 4.11. Pokušaj čitanja poruke prilikom LSB napada uništenja

Usporedba histograma originalne slike i ključ slike pokazivala je veliku sličnost – oko 99% za male poruke, 88% za poruke veličine 60% originalne te oko 69% za postotak približan 100% originalne (slika 4.12.).

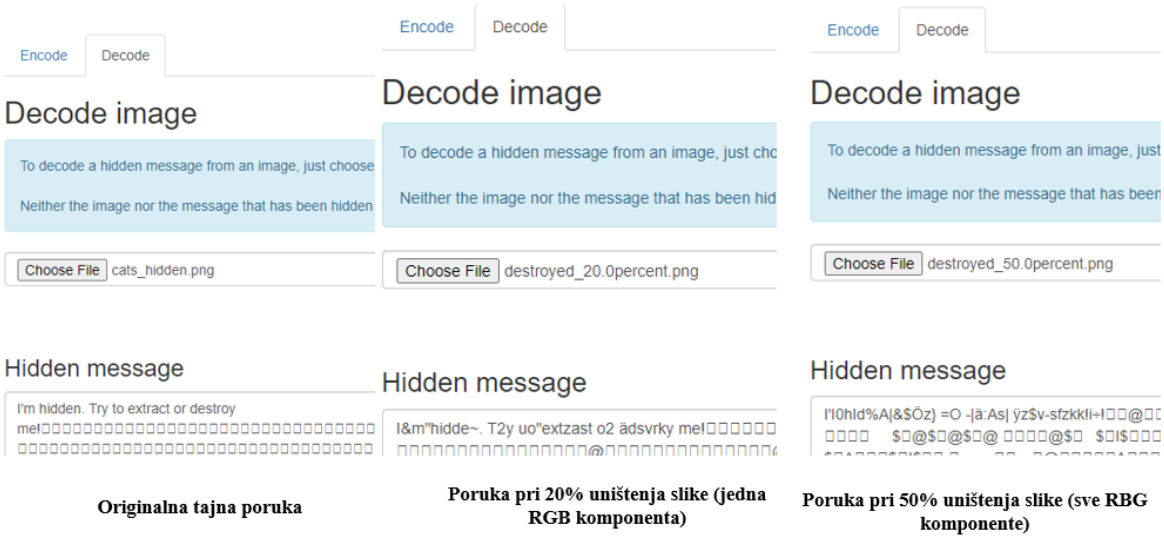


Postotak sličnosti: 88.6236% **Postotak sličnosti: 69.5071%**

Slika 4.12. Usporedba postotka sličnosti histograma pri različitim postotcima ugradnje

Daljnja testiranja provedena su uz online alat *Online Steganography* [54]. Razmatrana je mogućnost čitanja i uništenja poruke skrivene unutar stego-objekta stvorenog od navedenog alata. Na slici 4.13. prikazana je usporedba pokušaja čitanja poruke preko istog alata, uz određeni postotak uništenja, tj. uz proveden napad iz aplikacije. Odabirom jedne RGB komponente poruka je djelomično uništena, dok je odabirom svih uništena u potpunosti. Osim toga, uspješnost uništenja ovisila je i o metodi provođenja, čime se sekvencijalna metoda pokazala korisnijom. Navedeno je dovelo do zaključka da bi navedeni alat mogao ugrađivati bitove sekvencijalno, što bi moglo omogućiti iščitavanje iz implementirane aplikacije. Testiranjem su odabrane sve RGB komponente, te je iščitavanje bilo uspješno. Jedini parametar koji utječe na (ne)točnost dobivene poruke je postotak koji se želi dekodirati, no odabirom 100% ovaj problem se u potpunosti rješava.

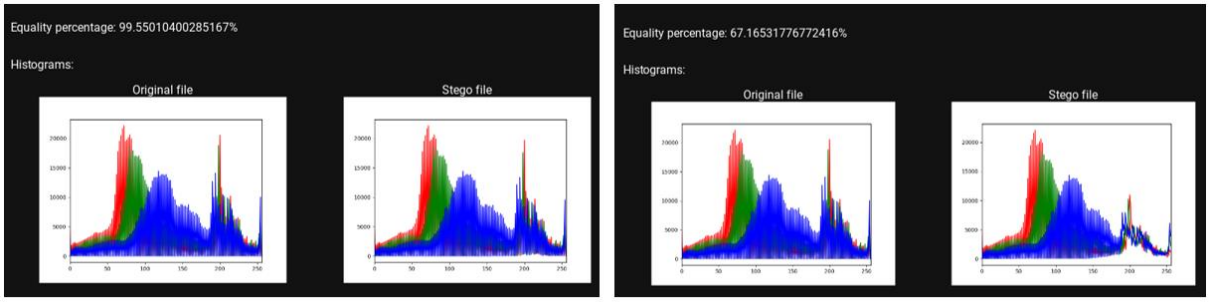
Zaključeno je kako navedeni alat koristi sekvencijalnu LSB tehniku ugradnje unutar svih RGB komponenti, što predstavlja jednu od opisanih varijanti ugradnje unutar aplikacije.



Originalna tajna poruka | **Poruka pri 20% uništenja slike (jedna RGB komponenta)** | **Poruka pri 50% uništenja slike (sve RBG komponente)**

Slika 4.13. Pokušaj dekodiranja iz alata *Online Steganography*, različiti postotci uništenja, postepeni način rada

S druge strane, ugradnja preko navedenog alata pokazala se otpornijom na analitičku metodu usporedbe histograma nasuprot načinu ugradnje preko aplikacije. Na slici 4.14. može se vidjeti postotak sličnosti prilikom ugradnje veće od 70%. S obzirom da je za alat *Online Steganography* sličnost puno veća, može se zaključiti kako je ugradnja otpornija.



Alat *Online Steganography* | **Aplikacija**

Slika 4.14. Usporedba sličnosti histograma alata *Online steganography* i aplikacije, ugradnja veća od 70%

4.4. Osvrt i mogućnosti poboljšanja

Napadima na LSB tehnike utvrđeni su optimalni parametri za tehniku kodiranja iz aplikacije – postepeni način uz sve RGB komponente. Ovime je osigurana otpornost na analitičku metodu usporedbe histograma. Ujedno je za dekodiranje potreban točan postotak (korak) te pristup korištenom algoritmu. Uništenje je uspješno za sve testirane alate koji koriste LSB-ove prilikom ugradnje, neovisno o algoritmu. BPCS tehnike analize pokazale su se praktičnim pri sistematskoj analizi, ujedno pružajući informacije gdje bi poruka mogla biti skrivena. Ukoliko se slika nalazi unutar bitnih razina, moguće je njeno dekodiranje. Tehnika metapodataka pruža lak i naizgled siguran način ugradnje, s time da treba pripaziti unutar kojih oznaka (engl. *tags*) će se poruka sakriti. S druge strane, dekodiranje je vrlo lako – bilo koji drugi alat koji omogućuje manipuliranje metapodacima može lako vidjeti gdje se nalazi poruka. Uništenja poruka su puno jednostavnija od provođenja analiza i pokušaja dekodiranja.

Struktura aplikacije omogućuje lako proširenje s novim napadima, čije bi dodavanje predstavljalo najbolje poboljšanje. Tehnika je mnogo, stoga je potrebno implementirati više napada kako bi mogućnosti bilo više, a time i vjerojatnost dekodiranja (ili uništenja) veća. Bilo bi povoljno dodati i testirati više opcija datoteka (testirano za jpg, bmp, png formate) te tipova (de)kodiranja znakova (utf-8, GBK, ISO-8859-1). Potrebni su specifični napadi na BPCS tehnike ugradnje. Dodatno, moguća su temeljitija testiranja anomalija koje se pojavljuju pri usporedbi histograma, poboljšanja sučelja te brzine izvođenja.

5. ZAKLJUČAK

Bez obzira na postojanje velikog broja načina kriptiranja i zaštite osobnih podataka i informacija, pojedinci koji žele uvijek mogu pronaći „vanjski“ način krađe, izmjene i nanošenja štete. Steganografija kao alternativna tehnika još je relativno nepoznata u današnjem svijetu, što ukazuje na opasnost korištenja u zloćudne svrhe, pogotovo ako žrtva toga nije svjesna. S obzirom na veliku povezanost i digitalizaciju, ima veliko područje primjene, bilo unutar organizacija ili radi osobne dobiti. S druge strane, treba biti svjestan mogućih ograničenja. Napadima na ove sustave – njihovim implementiranjem i provođenjem, osigurava se način provjere otpornosti korištenih tehnika, kao i mogućnost obrane od zloćudne upotrebe.

Istraživačkim dijelom utvrđena je opasnost i korisnost steganografskih tehnika, kao i nepouzdanost postojećih napada, prvotno zbog male količine i neraznolikosti. Implementirana je aplikacija koja omogućuje provođenje ukupno 8 napada, od kojih su 3 zasnovana na LSB steganografiji, 3 na BPCS steganografiji te 2 na steganografiji metapodataka. Dodatno su omogućene ugradnje preko LSB steganografije i steganografije metapodataka. Implementirani napadi omogućili su provođenje varijabilnog načina rada, te je time osigurana proširena primjena i sistematičnost. Testiranjem ove varijabilnosti uspješno su uništene sve poruke temeljene na LSB steganografiji, što je potvrdilo njihovu laku degradaciju. Osim toga, LSB napadi su uspjeli uništiti i poruke skrivene preko alata *SSuite Picsel*. Izvlačenja poruke preko LSB napada pokazala su se uspješnim, osim iz aplikacije, i kod alata *Online Steganography*. Ugradnja je bila jednostavna, no omogućavala je veliku slobodu pri modifikaciji toka, čime se može postići veća otpornost. Detekcija te dekodiranje slike skrivene BPCS tehnikom bilo je uspješno, dok uništenje nije. Iščitavanje, ugradnja i uništenje metapodataka bilo je jednostavno i lako provedivo, iz čega slijedi kako je tehnika slabo otporna. Analitički napad usporedbe histograma pokazao se uspješnim samo za pojedine alate, uz veliku ovisnost o veličini poruke koja se želi ugraditi (što je veća poruka, rezultati su vjerodostojniji). Zaključeno je kako napadi mogu biti pouzdani ukoliko korišten alat omogućuje provedbu i varijacije više njih.

Za veće područje primjene bilo bi potrebno implementirati više napada. Dodatni analitički napadi poput *Chi-Square Attack* i RS napada pružili bi veću kontrolu i bolje rezultate. Nadalje, korisna bi bila proširenja postojećih napada poput omogućavanja kontrole RGB komponenata pri preklapanju bitnih razina, većih formata čitanja/ spremanja te odabira korelacijske metode pri usporedbi histograma.

LITERATURA

[1] J. Bulao, How Much Data Is Created Every Day in 2022? [online], TechJury, 2022., dostupno na: https://techjury.net/blog/how-much-data-is-created-every-day/ [17.08.2022.]

[2] J. Wise, How much data is created every day in 2022? [online], Earthweb, 2022., dostupno na: https://earthweb.com/how-much-data-is-created-every-day/ [17.08.2022.]

[3] Y. Shi, Data Security and Privacy Protection in Public Cloud, IEEE International Conference on Big Data (Big Data), str. 4812-4819, Seattle, WA, USA, prosinac 2018.

[4] J. Qin, Y. Luo, X. Xiang, Y. Tan, i H. Huang, Coverless Image Steganography: A Survey, IEEE Access, sv. 7, str. 171372-171394, 2019.

[5] S. Wiseman, Stegware - Using Steganography for Malicious Purposes, prosinac 2017.

[6] L. Li, W. Zhang, K. Chen, i N. Yu, Steganographic Security Analysis From Side Channel Steganalysis and Its Complementary Attacks, IEEE Transactions on Multimedia, sv. 22, izd. 10, str. 2526-2536, prosinac 2019.

[7] Y. JinaChanu, K. Manglem Singh, i T. Tuithung, Image Steganography and Steganalysis: A Survey, International Journal of Computer Applications, sv. 52, izd. 2, str. 1-11, kolovoz 2012.

[8] T. Morkel, J. H. P. Eloff, i M. S. Olivier, An overview of image steganography, ISSA, sv. 1, izd. 2, str. 1-11, 2005.

[9] R. Gibson, Steganography: Hiding Data In Plain Sight [online], UNC, College of Arts and Sciences, Computer Science, dostupno na: https://www.cs.unc.edu/~lin/COMP089H/LEC/steganography.pdf [17.08.2022.]

[10] H.-J. Zepernick, D. Tran, i T. Chu, LSB Data Hiding in Digital Media: A Survey, EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, travanj 2022.

[11] C. McNamara, Field Guide to Consulting and Organizational Development: What Is an Open System?, Authenticity Consulting, LLC, str. 143-145, MN, USA, 2017.

[12] S. H. Abdullah, Steganography Methods and some application (The hidden Secret data in Image), Tikrit Journal of Pure Science, sv. 15, izd. 2, str. 105-111, 2010.

[13] J. Ashok, Y. Raju, S. Munishankaraiah, i K. Srinivas, Steganography: An Overview, International Journal of Engineering, Science and Technology, sv. 2, izd. 10, str. 5985-5992, listopad 2010.

[14] N. F. Johnson, Z. Duric, i S. Jajodia, Information Hiding: Steganography and Watermarking - Attacks and Countermeasures, izd. 1, Springer Science & Business Media, 2001.

[15] F. A. P. Petitcolas, R. J. Anderson, i M. G. Kuhn, Information hiding - a survey, Proceedings of the IEEE, sv. 87, izd. 7, str. 1062-1078, srpanj 1999.

[16] Jinyuan Tao, Sheng Li, Xinpeng Zhang, i Zichi Wang, Towards Robust Image Steganography, IEEE Transactions on Circuits and Systems for Video Technology., sv. 29, izd. 2, str. 594-600, veljača 2019.

[17] G. P. Rajkumar, V. S. Malemath, i M.Tech Student, Department of CSE, KLE Dr M S Sheshgiri College of Engineering & Technology, Udyambag Belgavi, India, Video Steganography: Secure Data Hiding Technique, Int. J. Comput. Netw. Inf. Secur., sv. 9, izd. 9, str. 38-45, rujan 2017.

[18] McAfee, Protecting Against Steganographic Threats [online], McAfee, lipanj 2017., dostupno na: https://www.mcafee.com/enterprise/en-us/assets/solution-briefs/sb-quarterly-threats-jun-2017-2.pdf [17.08.2022.]

[19] A. Shulmin i E. Krylova, Steganography in contemporary cyberattacks, SecureList, 2017., dostupno na: https://securelist.com/steganography-in-contemporary-cyberattacks/79276/?ref=hackernoon.com [17.08.2022.]

[20] C. P. Sumathi, T. Santanam, i G. Umamaheswari, A study of various steganographic techniques used for information hiding, International Journal of Computer Science and Engineering Survey, sv. 4, izd. 6, str. 9-25, prosinac 2013.

[21] R. Crandall, Some Notes on Steganography, Steganography Mailing List, prosinac 1998.

[22] A. W. A. Wahab, M. Hussain, Y. I. B. Idris, A. T. S. Ho, i K.-H. Jung, Image steganography in spatial domain: A survey, Signal Processing: Image Communication, sv. 65, str. 46-66, ožujak 2018.

[23] D.-C. Wu i W.-H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recognition Letters, sv. 24, str. 1613-1626, 2003.

[24] J. Liu, T. Zhou, Z. Zhang, Y. Ke, Y. Lei, M. Zhang, X. Yang, Digital Cardan Grille: A Modern Approach for Information Hiding, Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, str. 441-446, 2018.

[25] J. Liu, T. Zhou, Z. Zhang, Y. Ke, Y. Lei, M. Zhang, X. Yang, Recent Advances of Image Steganography With Generative Adversarial Networks, IEEE Access, sv. 8, str. 60575-60597, 2020.

[26] R. Vijaya Kumar Reddy, K. Prudvi Raju, L. Ravi Kumar, i M. Jogendra Kumar, Grey level to RGB using YCbCr color space Technique, International Journal of Computer Applications, sv. 147, izd. 7, str. 25-28, kolovoz 2016.

[27] FarhanMax, RGB vs YCbCr: What are the Main differences [online], 10Scopes, dostupno na: https://10scopes.com/rgb-vs-ycbcr/ [17.08.2022.]

[28] A. Srinivasan, S. Kolli, i J. Wu, Steganographic information hiding that exploits a novel file system vulnerability, International Journal of Security and Networks, sv. 8, izd. 2, str. 82-93, 2013.

[29] J. Lubacz, W. Mazurczyk, i K. Szczypiorski, Principles and Overview of Network Steganography, IEEE Communications Magazine, sv. 52, izd. 5, str 225-229, 2014.

[30] S. S. Khaire i S. L. Nalbalwar, Review: Steganography - Bit Plane Complexity Segmentation (BPCS) Technique, International Journal of Engineering, Science and Technology, sv. 2, izd. 9, str. 4860-4868, rujan 2010.

[31] L. Y. Por, K. Wong, i K. O. Chee, UniSpaCh: A text-based data hiding method using Unicode space characters, Journal of Systems and Software, sv. 85, izd. 5, str. 1075-1082, svibanj 2012.

[32] I. Banerjee, S. Bhattacharyya, i G. Sanyal, Study and Analysis of Text Steganography Tools, International Journal of Computer Network and Information Security, sv. 5, izd. 12, str. 45-52, studeni 2013.

[33] P. Shankdhar, Best tools to perform steganography [online], Infosec Institute, 2020., dostupno na: https://resources.infosecinstitute.com/topic/steganography-and-tools-to-perform-steganography/ [17.08.2022.]

[34] Nakasoft, Xiao Steganography, Softonic, dostupno na: https://xiao-steganography.en.softonic.com/ [17.08.2022.]

[35] S. Vaidya, OpenStego [online], OpenStego, dostupno na: https://www.openstego.com/ [17.08.2022]

[36] A. Kumar, 35 Best Free Steganography Software For Windows [online], ListOfFreeware, dostupno na: https://listoffreeware.com/list-of-best-free-steganography-software-for-windows/ [17.08.2022.]

[37] K. Abhiram, SteganPEG [online], Apponic, dostupno na: https://steganpeg.apponic.com/ [17.08.2022.]

[38] MRP-IM, Hide'N'Send [online], Download.com, A Red Ventures Company, dostupno na: https://download.cnet.com/Hide-N-Send/3000-2092_4-75728348.html [17.08.2022.]

[39] Don Caeiro i Sanjana S, Detection of Steganography using Metadata in Jpeg Files, International Journal of Forensic Computer Science, sv. 10, izd. 1, str. 23-28, prosinac 2015.

[40] S. A. Laskar i K. Hemachandran, A Review on Image Steganalysis techniques for attacking Steganography, International Journal of Engineering Research and Technology, sv. 3, izd. 1, str. 3400-3410, siječanj 2014.

[41] Andreas Westfeld i Andreas Pfitzmann, Attacks on Steganographic Systems, International workshop on information hiding, str. 61-76, Springer, Berlin, Heidelberg, 1999.

[42] Metadata for Data Management: A Tutorial: Definition [online], UNC University Libraries, svibanj 2022., dostupno na: https://guides.lib.unc.edu/metadata/definition [17.08.2022.]

[43] S. Choudhury, A. P.P, i M. Sethumadhavan, Stegware Destruction Using Showering Methods, International Journal of Innovative Technology and Exploring Engineering, sv. 8, izd. 6S3, str. 256-259, travanj 2019.

[44] F. Lundth, Pillow (PIL Fork) 9.2.0 documentation [online], Alex Clark and Contributors, dostupno na: https://pillow.readthedocs.io/en/stable/handbook/overview.html [17.08.2022.]

[45] Matplotlib: Visualization with Python [online], The Matplotlib development team, dostupno na: https://matplotlib.org/ [17.08.2022.]

[46] Multiprocessing - Process-based parallelism [online], Python, dostupno na: https://docs.python.org/3/library/multiprocessing.html [17.08.2022.]

[47] GeeksForGeeks, OpenCV Python Tutorial [online], GeeksForGeeks, dostupno na: https://www.geeksforgeeks.org/opencv-python-tutorial/#images [17.08.2022.]

[48] W3schools, NumPy Introduction [online], W3schools, dostupno na: https://www.w3schools.com/python/numpy/numpy_intro.asp [17.08.2022.]

[49] Exif2, IPTC & XMP metadata and ICC Profile [online], Exif2 dostupno na: https://exiv2.org/ [17.08.2022.]

[50] Python, Os - Miscellaneous operating system interfaces [online], Python, dostupno na: https://docs.python.org/3/library/os.html [17.08.2022.]

[51] OpenCv, Histogram Comparison [online], OpenCV, dostupno na: https://docs.opencv.org/3.4/d8/dc8/tutorial_histogram_comparison.html [29.08.2022.]

[51] Python image manipulation using PIL(LSB) [online], StackOverflow, 2019., dostupno na: https://stackoverflow.com/questions/58194992/python-image-manipulation-using-pillsb [17.08.2022.]

[52] SSuite Picsel Security [online], Cnet, 2018., dostupno na: https://download.cnet.com/SSuite-Picsel-Security/3000-2092_4-75992724.html [17.08.2022.]

[53] Stylesuxx, Steganography online [online], Stylesuxx, dostupno na: https://stylesuxx.github.io/steganography/ [17.08.2022.]

SAŽETAK

U teorijskom dijelu rada opisan je pojam steganografije, kontekst unutar današnjeg digitalnog doba te načini i slučajevi implementacije. Opisani su mogući napadi na ovakve sustave te važnost njihova provođenja. Steganografija pruža alternativu naspram kriptografije, ali s određenim manama i prednostima. Potrebno je vidjeti koje metode implementacije su sigurne, u kojoj mjeri, te istražiti moguće napade. Napadi trebaju pružiti povratnu informaciju o otpornosti – bilo radi testiranja sigurnosti komunikacije ili pokušaja uništenja zloćudnog sadržaja. Činjenica je da je u današnjem svijetu punog komunikacije teško kontrolirati bilo kakav promet, no napadi bi trebali biti poznati da, u slučaju da postoji sumnjiva komunikacija ili zloćudne datoteke, moguće provesti odgovarajuće radnje. Praktični dio rada predstavlja aplikaciju koja služi kao „omotač“ za dodavanje novih napada te demonstraciju nekih od poznatijih tehnika. Najjednostavnije je bilo provesti napade uništenja preko manipulacije najmanje značajnih bitova. Dekodiranje je moguće, ali izrazito specifično i zahtjevno. Potrebno je pružiti izbor između različitih metoda i varijacija kako bi napadi bili uspješni.

Ključne riječi: napadi na steganografske sustave, originalna datoteka, steganografija, stego-objekt, tajna poruka

ABSTRACT

Attacks on steganographic systems

The theoretical part of the paper describes the concept of steganography, its context in today's digital age, and the methods and cases of implementation. Possible attacks on such systems are described, as well as their importance. Steganography provide an alternative to cryptography, but with certain advantages and disadvantages. It's necessary too see which encoding methods are safe, to what extent, and to investigate possible attacks. Attacks should provide feedback on their resilience – either to test the safety of the communication or to try to destroy malicious content. The fact is that in today's world communication is difficult to control, but attacks should be known so that, in case there is in fact a suspicious communication taking place or malicious files present, possible to take appropriate actions. The practical part of the work presents an application that serves as a „wrapper“ for adding new attacks, as well as demonstrating some of the more well-known techniques. LSB destruction was the simplest to carry out. Decoding is possible, but extremely specific and demanding. Different methods and variations need to be provided for attacks to be successful.

Keywords: attacks on steganographic systems, original file, steganography, stego-object, secret message

ŽIVOTOPIS

Gabriel Veselovac, rođen je 28.03.2000. godine u Osijeku, s prebivalištem u Podravskim Podgajcima. Nakon završene opće gimnazije i osnovne glazbene škole u Donjem Miholjcu, upisuje se na Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku na Preddiplomski sveučilišni studij Računarstvo 2019. godine.

Dobitnik je STEM stipendije radi izvrsnosti tokom studiranja. Trenutno je zaposlen kao junior razvojni programer u tvrtki „Span“ unutar odjela „Software Applications Department“. Radio je kao demonstrator na kolegiju Digitalna elektronika. Sudjelovao je na „SmartSoc – Education of Future ICT Experts Based on Smart Society Needs“ projektu, brojnim natjecanjima te edukacijskim radionicama.

_____

Potpis autora

PRILOZI

1. „Napadi na steganografske sustave“ u .docx formatu
2. „Napadi na steganografske sustave“ u .pdf formatu
3. Izvorni kod