SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 58

Steganografija i steganaliza digitalnih slika

Goran Radanović

Zagreb, lipanj 2010.

Steganografija i steganaliza digitalnih slika

Sažetak

Steganografija je znanstvena disciplina koja proučava metode skrivanja tajne poruke unutar medija bezazlenog sadržaja. Steganaliza je znanstvena disciplina čiji je osnovni cilj detekcija poruke skrivene steganografskim metodama. Rad daje uvid u steganografske tehnike supstitucije i transformacije domene te tehnike statističke steganalize.

Ključne riječi: *Steganografija, Steganaliza, Objekt nositelj, Stego objekt, LSB, Hi kvadrat test, RS steganaliza, JPEG, JSTEG, OutGuess, F5*

Image steganography and steganalysis

Abstract

Steganography is a scientific discipline that studies methods of hiding secret messages within the media with harmless content. Steganalysis is a scientific discipline whose primary goal is detection of messages hidden with steganographic methods. The paper provides insight into steganographic techniques of substitution and domain transformation, and statistical steganalysis.

Key words: *Steganography, Steganalysis, Cover object, Stego object, LSB, Chi square test, RS steganalysis, JPEG, JSTEG, OutGuess, F5*

Sadržaj

1. Uvod

Sve veći broj korisnika Interneta uzrokuje povećani interes za područje sigurnosti računalnih sustava. Postavlja se zahtjev za sprječavanjem neovlaštene upotrebe i promjene te uništavanja distribuiranih informacija. Posebice je to bitno za vojne i bankovne informacijske sustave. Jedan vid zaštite informacija je i očuvanje njihove tajnosti.

Tajnost informacija se može sačuvati njihovim kriptiranjem. Korištenjem matematičkih alata moguće je postići da samo ovlaštene osobe imaju pristup povjerljivim podacima. Kriptiranjem se izvorna poruka ispremiješa tako da postaje nerazumljiva, a dobivanje izvorne informacije iz kriptirane poruke se postiže dekriptiranjem uz korištenje odgovarajućeg ključa. Iako treća strana ne može odgonetnuti tajnu poruku, ona je svjesna njenog postojanja. Drugi način očuvanja tajnosti informacija je skrivanje poruke u objekt koji već nosi nekakvu informaciju razumljivog sadržaja. Sada treća strana nije niti svjesna postojanja tajne poruke. Ovakav vid očuvanja tajnosti informacija zadire u područje steganografije. Uz područje steganografije se prirodno veže i područje steganalize čiji je primarni cilj detekcija skrivene poruke, a zatim i njeno izdvajanje iz objekta nositelja poruke.

U radu se opisuju postupci i tehnike steganografije i steganalize digitalnih slika. Naglasak je stavljen na BMP i JPEG formate slika. U prvom dijelu su dani osnovni pojmovi steganografije i steganalize te njihova kategorizacija. Drugi dio rada podrobnije ulazi u opis metoda steganografije zasnovanih na supstituciji i transformaciji domene te pripadajućih tehnika statističke steganalize. Prikazuju se rezultati uspješnosti napada na pojedine steganografske algoritame. Rad završava pregledom implementiranih tehnika u razvojnom okruženju MATLAB.

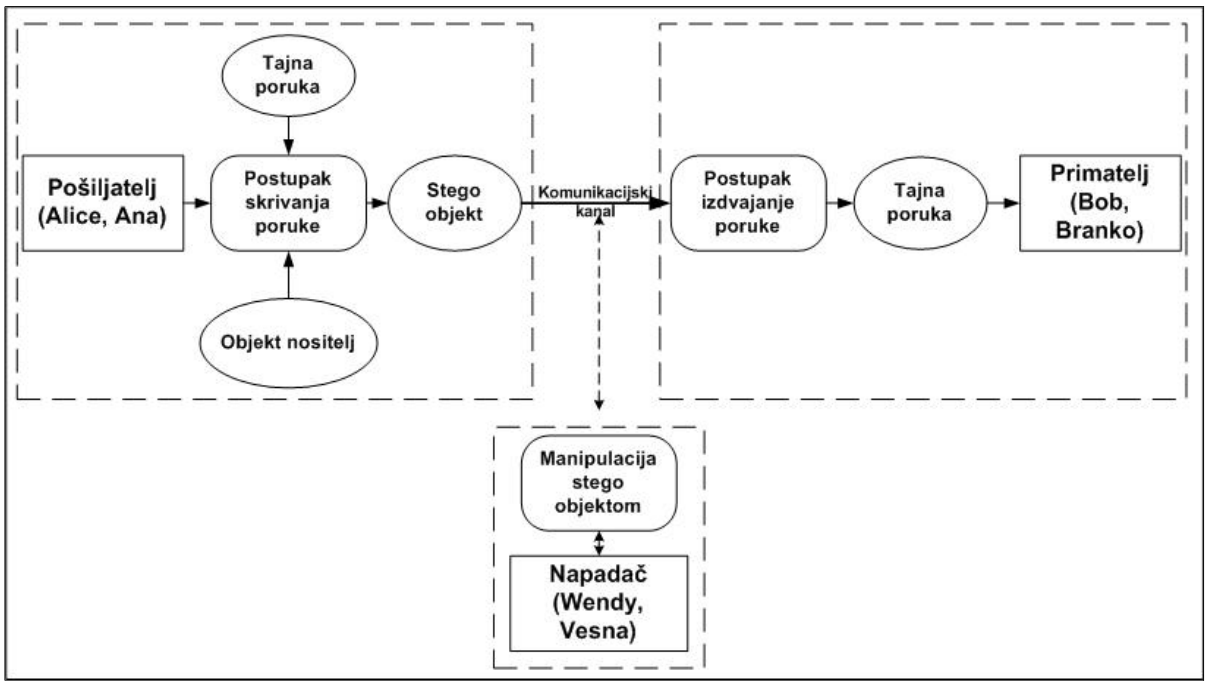
2. Steganografija i steganaliza

2.1 Steganografija

Steganografija (*Steganography*) je znanost i vještina zapisivanja poruke na takav način da nitko, osim pošiljatelja i primatelja poruke, ne posumnja u njezino postojanje [4]. Riječ steganografija je proizašla iz grčkih riječi *steganos* i *graphein*, što u prijevodu znači *skriveno pisanje* [3]. Steganografija spada u područje skrivanja informacija (*Information Hiding*).

2.1.1 Steganografski sustav

Osnovni principi steganografije se mogu predočiti jednostavnim primjerom zvanim *problem zatvorenika* [1], [2]. Neka su Ana (Alice) i Branko (Bob) zatvorenici koji planiraju bijeg iz zatvora i neka je Vesna (Warden) upravitelj zatvora. Nadalje, neka se Ana i Branko nalaze u različitim dijelovima zatvora tako da mogu komunicirati samo pisanim porukama koje kontrolira Vesna. Kako bi se dogovorili o bijegu, Ana i Branko moraju slati poruke koje nisu sumnjivog sadržaja, tj. trebaju koristiti tehnike steganografije. Vesna, koja kontrolira izmjenu poruka, može pri tome pokušati odgonetnuti postoji li kakav sumnjivi sadržaj u njihovim porukama, ali i, namjerno ili nenamjerno, mijenjati sadržaje njihovih poruka. Sam pokušaj otkrivanja sumnjivog sadržaja predstavlja pasivni napad (*passive attack*), dok mijenjanje sadržaja poruke predstavlja aktivni napad (*active attack*). Vesna može i krivotvorenu poruku predati Ani ili Branku. Takav napad se zove zlonamjeran napad (*malicious attack*) [2]. Sada se može definirati steganografski sustav.



Slika 2.1 Shematski prikaz steganografskog sustava

Steganografski sustav je prikazan na slici 2.1. Pošiljatelj (Alice, Ana) šalje primatelju (Bob, Branko) tajnu poruku. Pošiljatelj postupkom skrivanja poruke umeće tajnu poruku u objekt nositelj (*cover object*) i time se dobije *stego* objekt. Stego objekt se prenosi komunikacijskim kanalom do primatelja. Primatelj postupkom izdvajanja poruke dobiva tajnu poruku. U

shematskom prikazu postoji i treća osoba zvana napadač (Wendy, Vesna). Napadač ima mogućnost manipulacije stego objektom. On može biti pasivan i samo prisluškivati komunikacijski kanal ili aktivan i, uz prisluškivanje kanala, modificirati stego objekt [1]. Također, napadač može i krivotvorene poruke slati sudionicima komunikacije. Cilj napadača je otkriti sadrži li objekt koji se šalje komunikacijskim kanalom skrivene informacije. U slučaju da napadač otkrije postojanje tajne poruke, on ju može pokušati izdvojiti ili narušiti stego objekt tako da se izgubi tajna poruka.

Postupak skrivanja, odnosno izdvajanja poruke, može biti tajan ili javan. Ukoliko je postupak javan, često se koristi dodatni ključ koji određuje točan slijed umetanja poruke. Taj ključ se razmijeni sigurnim kanalom. Time se izbjegava da napadač jednostavno dođe do skrivene poruke. Osim toga se sama poruka može kriptirati nekim kriptografskim algoritmom čime se postiže dodatna sigurnost u očuvanju tajnosti poruke. Bitno je istaknuti da je osnovni cilj napadača detektirati skrivenu poruku, a ne ju nužno i pročitati. Naime, pretpostavlja se da napadač ima mogućnost manipulacije stego objektom. Stoga on može narušiti komunikaciju primatelja i pošiljatelja i samom izmjenom stego objekta.

Koliko je neki steganografski sustav siguran ovisi o tome koliko se dobro odupire pasivnim, aktivnim i zlonamjernim napadima. Steganografski sustav je robustan ukoliko se skrivena informacija može izmijeniti tek "većim" izmjenama stego objekta [1]. Siguran steganografski sustav ispunjava četiri uvjeta [1]:

- Algoritam skrivanja je javan, ali koristi se tajni ključ.
- Samo onaj tko posjeduje tajni ključ može detektirati, izvaditi i dokazati postojanost tajne poruke. Nitko drugi ne može otkriti nikakav statistički trag o postojanju tajne poruke.
- Čak i kada napadač poznaje sadržaj jedne prenesene poruke, mala je vjerojatnost da će odgonetnuti sadržaj preostalih poruka.
- Detekcija tajne poruke je računalno prezahtjevna.

Iako steganografski sustav koji ima javni algoritam skrivanja, a ne koristi tajni ključ, nije siguran, ponekad je on zadovoljavajući. To se odnosi na sustave koji raspolažu s ogromnom količinom informacija pa se ne mogu analizirati svi objekti nositelji informacija. U tom slučaju nepostojanje ključa može biti i olakotna okolnost s obzirom da nije potrebno prenijeti tajni ključ između sudionika komunikacije.

Važno je uočiti da se steganografija u bitnome razlikuje od kriptografije. Naime, kod kriptografije se tajnost poruke postiže modifikacijom poruke, dok se kod steganografije tajnost poruke postiže njezinim skrivanjem. Također, bitno je naglasiti da postoje razlike između digitalne steganografije i digitalnog vodenog žiga. Digitalna steganografija ima zadatak skrivenog prijenosa tajne poruke kroz digitalni medij, dok se digitalni vodeni pečat koristi za *označavanje* digitalnog signala.

2.1.2 Steganografija kroz povijest

Zanimljiv kronološki pregled steganografije kroz povijest je dan u [2], a ovdje su nabrojene neke zanimljivosti preuzete iz [2], [3] i [4]:

- U drevnoj Grčkoj su se za pisanje upotrebljavale drvene ploče prelivene voskom. Tajna poruka se mogla napisati na drvenu ploču prije nego se ploča prelila voskom. Nakon što se ploča prelila voskom, tajna poruka je ostala skrivena.

- Osim voštane ploče, tajna informacije se skrivala i na tijelima glasnika. Glava glasnika bi se obrijala i na njoj bi se tetovirala tajna poruka. Nakon što bi kosa narasla, tajna poruka bi ostala skrivena, a čitanje poruke bi se vršilo brijanjem glasnikove glave. Ovakve tehnike su korištene u Grčko-Perzijskim ratovima.
- Drevni Kinezi su na tanki komad svile zapisivali poruku, zatim su taj komad stavljali u voštanu kuglicu koju bi glasnik progutao.
- Redovnik Johannes Trithemius je oko 1500. napisao knjigu *Steganographia* u kojoj je opisao kako sakriti poruku unutar bezazlenog teksta. Knjiga je izdana 1606.
- Talijanski znanstvenik Giovanni Porta, rođen 1535. godine, je otkrio kako sakriti poruku unutar skuhanog jaja. Kao tintu je koristio mješavinu stipsa i octa. Poruku bi nanio na skuhano jaje, a tinta od stipsa i octa bi prodrla kroz ljusku jaja. Skrivena poruka bi se očitala s bjelanjka nakon što bi se jaje ogulilo.
- Tijekom Drugog svjetskog rata se nevidljiva tinta koristila za prijenos tajne poruke. Dobivanje skrivene poruke zasnivalo se na svojstvu da tinta postaje vidljiva prilikom zagrijavanja, odnosno kontakta s određenom kemijskom supstancom.
- Također, poruke su se znale prenositi u predivu korištenjem znakova Morseove abecede ili na poleđini poštanskih maraka.
- Tijekom i nakon Drugog svjetskog rata bila je česta uporaba mikrofotografija (*microdots*). Mikrofotografije su bile veličine točke pa su se lako mogle smjestiti, tj. nalijepiti, u tekst. Ipak, tako nalijepljene mikrofotografije bi reflektirale svjetlost kada bi se papir okrenuo pod određenim kutom pa su bile primjetne.
- Valja spomenuti i korištenje nulte šifre u špijunaži. Tako je Velvalee Dickinson, japanska špijunka i preprodavačica lutaka, slala pisma narudžbe u Južnu Ameriku u kojima su se nalazile skrivene informacije o kretanjima brodova.

2.1.3 Tipovi steganografije

Dva su osnovna tipa steganografije: tehnička steganografija (*technical steganography*) i lingvistička steganografija (*linguistic steganography*).

Tehnička steganografija se ogleda u korištenju posebnih tehničkih sprava, uređaja, instrumenata i metoda u skrivanju poruke. U tehničku steganografiju tako spada [2]:

- *Nevidljiva tinta*: Tinta koja nema boje dok se ne dovede u kontakt s toplinom ili nekom kemijskom supstancom.
- *Korištenje skrovitih mjesta*: Primjerice, skrivanje na dnu pivske bačve ili tetovaža ispod kose.
- *Mikrofotografije* (*microdots*): Fotografije veličine manje od pola milimetra.
- *Računalom zasnovane metode*: Mnogobrojne metode proizašle upotrebom računala.

Lingvistička steganografija obuhvaća steganografske metode koje skrivaju tajnu poruku u objekt nositelj nastojeći je prikazati kao bezazlen skup informacija [3]. Uobičajeno je objekt nositelj jezičnog sadržaja. Lingvističku steganografiju dijelimo na [3]:

- *Semagrami* (*semagrams*): Razmještaju se simboli, znakovi ili objekti, ali tako da promjene koje nose informaciju ne budu vidljive. Dvije su vrste:
    - *Vizualni semagrami* (*visual semagrams*): Fizički objekti i njihov razmještaj upućuju na sadržaj skrivene informacije.
    - *Tekstualni semagrami* (*textual semagrams*): Modifikacijom teksta se postiže skrivanje informacije (primjerice veličina slova, font i dr.).

- *Otvoreni kodovi* (*open codes*): Poruka s javnom informacijom se koristi kako bi se skrila tajna poruka. Otvorene kodove dijelimo na:
    - *Žargonski kod* (*jargon code*): Koristi se jezik koji razumije samo određena skupina ljudi.
    - *Skrivene šifre* (*covered ciphers*): Poruka se može izvaditi iz stego objekta samo ako je poznat algoritam skrivanja tajne poruke. Tako u skrivene šifre spadaju:
        - *Rešetkasta šifra* (*grille ciphers*): Koriste se maske s otvorima, tj. rupicama, i slova koja se pojavljuju u rupicama upućuju na skrivenu poruku.
        - *Nulta šifra* (*null cipher*): Skrivena poruka se čita na temelju prethodno definiranih jednostavnih pravila. Primjerice, jednostavno pravilo može biti da se čita svako deseto slovo.

2.1.4 Kategorizacija steganografskih tehnika

Steganografske tehnike se mogu podijeliti u sljedećih šest kategorija [1], [2]:

- *Tehnike supstitucije*: Suvišni dijelovi nositelja se iskorištavaju za umetanje tajne informacije. Najpoznatija tehnika ove kategorije je LSB (*Least Significant Bit*) koja bitove najmanje važnosti podešava tako da postanu nositelji skrivene poruke.
- *Tehnike transformacije domene*: Modifikacija se vrši u transformiranoj domeni. Najčešće se koriste diskretna kosinusna transformacija (*Discrete Cosine Transformation, DCT*) i diskretna Fourierova transformacija (*Discrete Fourier Transformation, DFT*).
- *Tehnike raspršenog slijeda* (*Spread Spectrum, SS*): Uski signal tajne poruke se nastoji sakriti unutar nositelja. Tajna poruka se modulira signalom šuma te se dodaje u nositelja: isključivo poznavanjem ključa je moguće iz naizgled slučajnog signala šuma dobiti skrivenu poruku. Dvije se metode raspršenog slijeda koriste u steganografiji: *tehnika raspršenog spektra direktnog slijeda* (*Direct Sequence Spread Spectrum, DSSS*) i *tehnika raspršenog spektra frekvencijskog skoka* (*Frequency Hopping Spread Spectrum, FHSS*).
- *Statističke metode*: Nositelj se podijeli na onoliko blokova kolika je veličina poruke. Svaki blok služi za skrivanje jednog bita tajne poruke. Ukoliko je bit poruke jednak 1, blok se modificira tako da primatelj može statističkim testiranjem hipoteze otkriti je li taj blok promijenjen. Ukoliko je bit poruke jednak 0, blok se ne mijenja.
- *Tehnike izobličavanja*: Tajna poruka se ne skriva direktno u nositelja, već se stvaraju preinake nositelja kako bi se prenijela tajna poruka. Zahtijeva se da primatelj poznaje originalnu verziju nositelja skrivene poruke.
- *Tehnike stvaranja nositelja skrivene informacije*: Tajna poruka se ne skriva u nositelju, već se na temelju nje stvara nositelj koji joj odgovara.

2.2 Steganaliza

Steganaliza (*Steganalysis*) je znanost i vještina detekcije poruke skrivene steganografskim metodama [4], [6]. Osnovni cilj steganalize je detekcija skrivene poruke, no u steganalizu spada i procjena veličine skrivene poruke, kao i njezino izdvajanje iz stego objekta. Tehnike steganalize predstavljaju napade na steganografske algoritme, odnosno metode analize steganografskih algoritama. Zbog toga se tim tehnikama testira koliko je steganografski

sustav siguran, tj. koliku količinu informacija je moguće sakriti u objekt nositelj prije nego se detektira skrivena poruka. Tako su neke steganografske metode tek otporne na vizualne napade, dok su druge otporne i na statističke testove, ali količina informacija koje mogu sigurno sakriti je znatno manja.

2.2.1 Kategorizacija tehnika steganalize

Tehnike steganalize se mogu podijeliti u sljedeće kategorije [5]:

- *Tehnike nadziranog učenja*: Iskorištavaju se tehnike nadziranog učenja kako bi se stvorio sustav koji je u mogućnosti obavljati steganalizu. Razlikuju se dvije faze oblikovanja takvog sustava: treniranje (učenje) sustava i eksploatacija (iskorištavanje) sustava.
- *Tehnike slijepe identifikacije*: Steganaliza se postavlja kao problem identifikacije sustava. Pri tome se iskorištavaju neka statistička svojstva slike nositelja i poruke koja se prenosi.
- *Tehnike statističke analize*: Problem steganalize se oblikuje kao statistički problem testiranja hipoteze. Postavi se statistički model slike nositelja i stego slike te se testira hipoteza o postojanju skrivenih podataka.
- *Hibridne tehnike*: Ove tehnike su mješavina prethodnih kategorija.

3. RC4 kriptografski algoritam

Često se podaci, prije nego se umetnu u objekt nositelj, kriptiraju. U radu je za kriptiranje korišten *RC4* algoritam koji, osim za kriptiranje, može poslužiti i kao generator pseudo-slučajnih brojeva. RC4 je najčešće korišten kriptografski algoritam toka podataka.

3.1 Generiranje podključeva

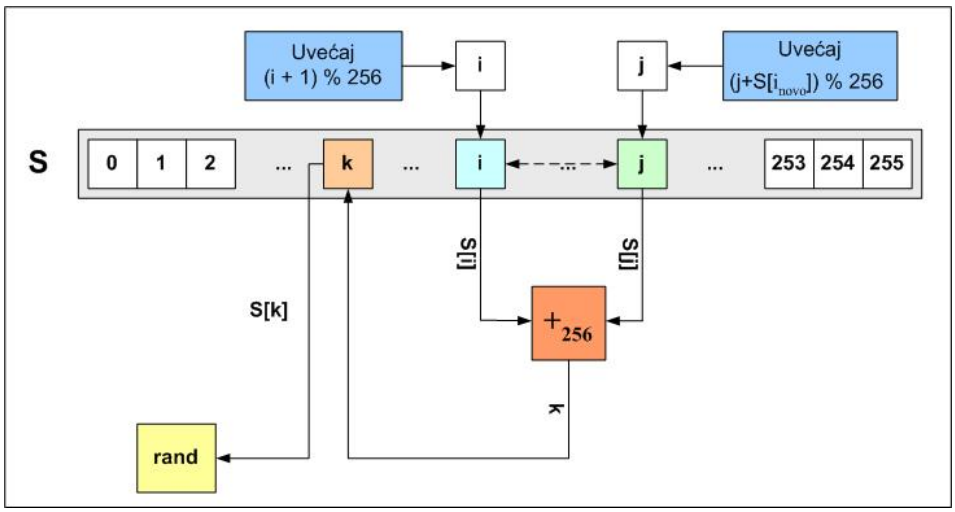
RC4 ima ključ koji može biti duljine od 1 do 256 okteta. Najčešće duljine ključa su između 5 i 40 okteta. Algoritam koristi 256 podključeva veličine 8 bita. Polje od 256 podključeva se naziva *S-box* [7]. Inicijalizacija podključeva se obavlja prema algoritmu prikazanom na slici 3.1.

```
RC4_inicijaliziraj(ključ){
    S[256] = 0;
    Za(i = 0; i < 256; i++) S[i] = i;
    Za(i = 0; i < 256; i++){
        j = (j + ključ[i % veličina(ključ)] + S[i]) % 256;
        Zamijeni(&S[i], &S[j]);
    }
    Vrati S;
}
```

Slika 3.1. Pseudokod inicijalizacije podključeva

3.2 Generiranje pseudo-slučajnih brojeva

Generiranje pseudo-slučajnih brojeva se kod RC4 algoritma obavlja prema shemi nacrtanoj na slici 3.2.



Slika 3.2. Generiranje pseudo-slučajnih brojeva

Dakle, brojači *i* i *j* se prvo uvećaju u aritmetici modulo 256: *i* se uveća za jedan, a *j* se uveća za vrijednost *S*[*i*] pri čemu se koristi vrijednost od uvećanog *i*. Nakon toga ti brojači pokazuju na brojeve u S kutiji koji međusobno zamjenjuju pozicije u S kutiji te se zbrajaju u aritmetici modulo 256. Rezultat zbrajanja je pokazivač *k* i on pokazuje na broj u S kutiji koji predstavlja izlaznu vrijednost. Početne vrijednosti brojača *i* i *j* su jednake 0.

3.3 Kriptiranje i dekriptiranje RC4 algoritmom

Kriptiranje i dekriptiranje RC4 algoritmom je identično. U osnovi se koristi gore opisani generator pseudo-slučajnih brojeva. Za ulazni podatak od 8 bita se generira 8-bitni pseudo-slučajni broj, a zatim se obavi XOR operacija tih dvaju brojeva. Pseudokod kriptiranja i dekriptiranja je prikazan na slici 3.3.

```
S = RC4_inicijaliziraj(ključ);
i = 0; j = 0;

Dok postoji podataka u ulaznom toku{
    i = (i + 1) % 256;
    j = (j + S[i]) % 256;
    Zamijeni(&S[i],&S[j]);
    k = (S[i] + S[j]) % 256;
    izlaz = ulaz XOR S[k];
}
```

Slika 3.3. Pseudokod kriptiranja i dekriptiranja RC4 algoritmom

4. Skrivanje u prostornoj domeni

Pregled steganografskih metoda i tehnika steganalize započinje s algoritmima koji operiraju u prostornoj domeni. U ovom poglavlju je dan naglasak na jednostavan LSB supstitucijski algoritam u prostornoj domeni te na metode statističke steganalize kojima se uspješno može napasti spomenuti steganografski algoritam.

4.1 Jednostavni LSB u prostornoj domeni

Kao što je već spomenuto, LSB supstitucijska metoda mijenja bitove najmanje važnosti tako da ih uskladi s tajnom porukom. Ovdje će se objasniti kako u slici 24-bitovnog BMP formata sakriti poruku koristeći LSB metodu.

U 24-bitovnom BMP formatu slike slikovni element je predstavljen kombinacijom crvene, zelene i plave boje. Drugim riječima, koristi se RGB sustav boja u kojem svaka od navedenih boja ima kanal veličine 8 bita. LSB metoda mijenja najmanje značajne bitove od svakog kanala. To znači da je moguće sakriti 3 bita po jednom slikovnom elementu. Konkretno, u slici veličine 256×256 slikovnih elemenata moguće je sakriti 196608 bita, tj. 24 KB.

Primjer rada LSB supstitucije je dan u tablici 4.1. Neka se skriva poruka 1010. Za svaki bit poruke se određuje u koji će slikovni element bit umetnut. Jedan način na koji se to može učiniti je slijedno prolaženje kroz sliku (primjerice od gornjeg lijevog ruba do donjeg desnog ruba). Ipak, najčešće se koristi generator pseudo-slučajnih brojeva (*Pseudo-Random Number Generator, PRNG*). U tom je slučaju bitno da pošiljatelj i primatelj koriste isto sjeme pri generiranju slučajnih brojeva kako bi primatelj mogao rekonstruirati skrivenu poruku. Sjeme generatora pseudo-slučajnih brojeva predstavlja steganografski ključ. Osim za inicijalizaciju PRNG-a, steganografski ključ se može koristiti i za kriptiranje same poruke. Neka je PRNG generirao pozicije: **(1, 2)**, **(3,1)**, **(3,3)** i **(2,3)**. Prvi bit poruke mijenja odgovarajući slikovni element jer se najmanje značajni bit slikovnog elementa razlikuje od bita poruke. Drugi bit poruke ne mijenja odgovarajući slikovni element jer se najmanje značajni bit slikovnog elementa ne razlikuje od bita poruke.

Tablica 4.1 Skrivanje poruke LSB metodom

Poruka koja se skriva: **1010**, *Polja generirana PRNG-om*: **(1, 2), (3,1), (3,3), (2,3)**		
100010001	10100010**1**	001011000
010010101	101001001	11101010**0**
11111000**0**	101000101	00011011**1**

Primatelj poruke, koristeći isto sjeme, odnosno ključ, može rekonstruirati originalnu poruku: prolazi kroz pozicije generirane generatorom slučajnih brojeva, čita zadnje bitove i tako dobiva poslanu poruku.

Pseudokod jednostavne LSB metode skrivanja (*simple LSB*, *sLSB*) implementirane u okviru rada je dan na slici 4.1. Ukoliko se algoritmu proslijedi steganografski ključ, tada se ključem inicijalizira PRNG, ali se i kriptira poruka koristeći RC4 algoritam. Ukoliko se ne zada ključ,

poruka se skriva slijedno bez kriptiranja. Algoritam prvo skriva veličinu poruke, a zatim i samu poruku.

```
LSB_sakrij(slika, poruka, ključ = -1){
    Ako(ključ == -1) pozicije = Slijedno();
    Inače{
        pozicije = PRNG(ključ);
        poruka = RC4(poruka, ključ);
    }
    bit_info = bitovi(poruka);
    vel_por = bitovi(duljina(bit_info)); //veličina je spremljena u 32 bita
    stego = slika;
    Za(i = 0; i < 32; i++){
        LSB(stego(pozicije[i])) = vel_por[i];
    }
    Za(i = 0; i < duljina(bit_info); i++){
        LSB(stego(pozicije[i + 32])) = bit_info[i];
    }
    Vrati stego;
}
```

Slika 4.1 Pseudokod skrivanja poruke LSB metodom

Na slici 4.2 je dan pseudokod dobivanja skrivene poruke iz stego slike. Opisani algoritam prvo dohvaća veličinu skrivene poruke, a potom i bitove skrivene poruke. Ukoliko je zadan ključ, algoritam dekriptira skrivenu poruku RC4 algoritmom.

```
LSB_dohvati(stego, ključ){
    Ako(ključ == -1) pozicije = Slijedno();
    Inače pozicije = PRNG(ključ);
    Za(i = 0; i < 32){
        vel_por[i] = LSB(stego(pozicije[i]));
    }
    duljina_poruke = int(vel_por);
    Za(i = 0; i < duljina_poruke; i++){
        bit_info[i] = LSB(stego(pozicije[i + 32]));
    }
    poruka = char(bit_info);
    Ako(ključ != -1) poruka = RC4(poruka, ključ);
    Vrati poruka;
}
```

Slika 4.2 Pseudokod dobivanja skrivene poruke (LSB metoda)

Na slici 4.3 je prikazan odnos originalne slike (slike nositelja) i stego slike. Skrivena poruka zauzima oko 90% kapaciteta namijenjenog za spremanje tajne poruke (90% zadnjih bitova).

Ovako opisana LSB metoda predstavlja brzu i robusnu tehniku skrivanja poruke u slici, no statističkim testovima je lako utvrditi postojanje skrivene poruke. Također, skrivena poruka može biti izgubljena prilikom manjih promjena slike poput onih koji se događaju u postupcima kompresije s gubitkom podataka.

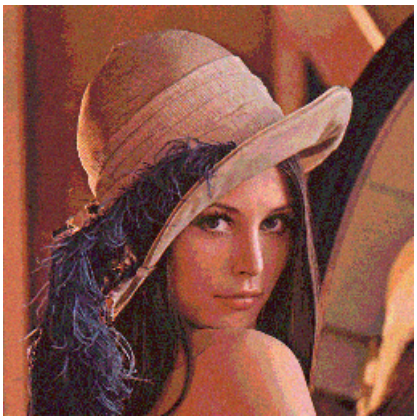
Osim zadnjeg bita, bita najmanje važnosti, LSB tehnika može koristiti nekoliko manje značajnih bitova. Zanimljivo je pogledati koliko je metoda robusna obzirom na vizualne napade, tj. koliko manje značajnih bitova se može koristiti, a da ljudsko oko ne primijeti veće nepravilnosti. Na slici 4.4 je prikazan utjecaj količine korištenih bitova na deformaciju slike. Ukoliko se koriste 3 bita najmanje važnosti, deformacije slike su neprimjetne ljudskom oku. Korištenjem većeg broja bitova deformacije na slici postaju primjetne ljudskom oku.

Robusnost na vizualne napade je bitna za sustave u kojima napadač ne može provjeravati sve slike zbog njihove brojnosti pa provjerava samo slike s vidljivim oštećenjima.

Slika 4.3 Originalna slika i stego slika (sLSB)

Originalna slika

3 bita najmanje važnosti se koriste za pohranu podataka



5 bitova najmanje važnosti se koriste za pohranu podataka

7 bitova najmanje važnosti se koriste za pohranu podataka

Slika 4.4 Robusnost LSB metode na vizualne napade

4.2 Skrivanje slike u slici

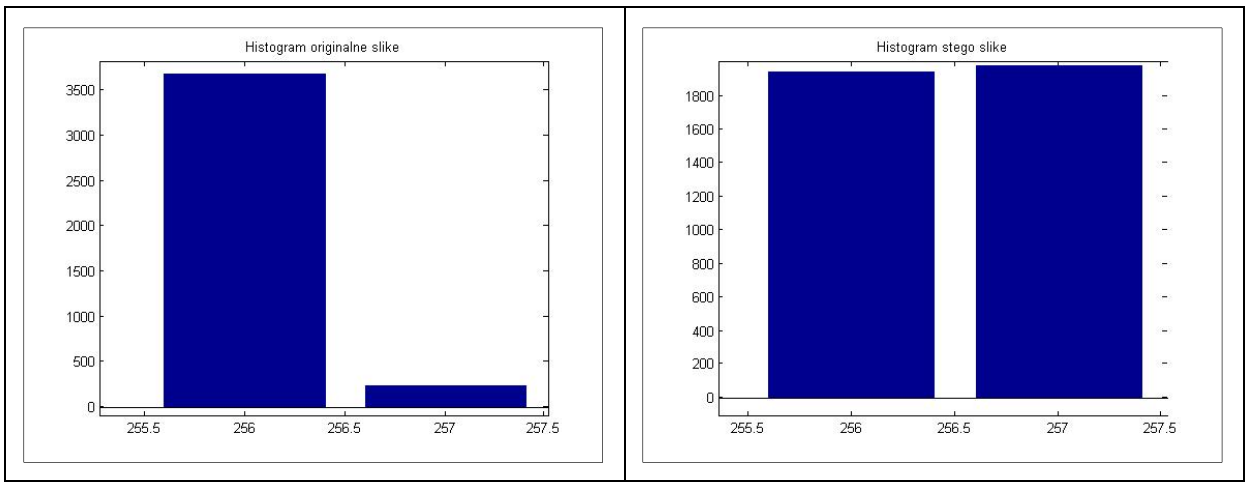
Skrivanje slike u slici je moguće postići *image downgrading* metodom. Metoda se temelji na gore opisanoj LSB metodi. U četiri najmanje značajna bita RGB kanala svakog slikovnog elementa nositelja se smještaju četiri najviše značajna bita RGB kanala odgovarajućeg slikovnog elementa tajne slike. Naime, bitovi veće važnosti više pridonose izgledu slike. Stoga je za očekivati da će tajna slika ostati neprimjetna ljudskom oku. Tajna slika se iz stego slike dobiva iz četiri manje značajna bita RGB kanala svakog slikovnog elementa. Valja uočiti da je ovim procesom dijelom izgubljen sadržaj tajne slike. Na slici 4.5 su prikazane slike u postupku *image downgrading*. U sliku nositelja se skriva tajna slika i time se dobiva stego slika. Iz stego slike se potom dohvaća skrivena slika.

	
Slika nositelj	*Tajna slika*
	
Stego slika	*Skrivena slika*

Slika 4.5 Skrivanje slike u slici

4.3 Steganaliza upotrebom hi kvadrat testa

Steganaliza upotrebom hi kvadrat testa spada u tehnike statističke analize. Hi kvadrat test se u statistici upotrebljava, između ostalog, i za određivanje razdiobe zadanih podataka. Ideja je, dakle, iskoristiti takav test kako bi se pokazalo ili opovrgnulo postojanje skrivene poruke uz neki nivo značajnosti. To se vrši promatranjem histograma slike, preciznije, promatranjem susjednih vrijednosti u histogramu slike. Ovakvi napadi su učinkoviti u otkrivanju postojanja skrivene poruke kada se algoritam umetanja temelji na LSB metodi.



Slika 4.6 Histogram originalne slike i stego slike za vrijednosti 256 i 257

LSB metoda umetanja mijenja najmanje značajan bit slikovnog elementa. Budući da takvo umetanje ne uzrokuje značajnije modifikacije slike, ljudskom oku takva promjena neće biti uočljiva. Ipak, takva promjena može značajno promijeniti neka statistička svojstva slike pa napadač može detektirati postojanje skrivene poruke. Jedna ideja detekcije se temelji na promatranju parova vrijednosti u histogramu slike: promatra se parna vrijednost i susjedna neparna vrijednost (za jedan veća od promatrane parne, primjerice 10 i 11). Uz pretpostavku da se bitovi skrivene poruke ravnaju prema uniformnoj razdiobi, tj. da se 1 i 0 u poruci pojavljuju jednako često, za očekivati je da će kod stego slike koja je značajno izmijenjena, tj. u koju je umetnuta velika količina podataka, iznosi odgovarajućih (susjednih) parnih i neparnih vrijednosti u histogramu slike biti otprilike jednaki. Kako to kod proizvoljne slike uobičajeno nije slučaj, približno jednaki iznosi parnih i odgovarajućih neparnih vrijednosti u histogramu predstavljaju indikator skrivene poruke. Na slici 4.6 su prikazane vrijednosti 256 i 257 u histogramu originalne slike i stego slike u koju je skrivena najveća moguća poruka. Iz slike se jasno vidi da umetanje poruke koja se ravna prema uniformnoj razdiobi ima tendenciju izjednačavanja susjednih razina u histogramu.

Za testiranje hipoteze o jednakosti odgovarajućih parnih i neparnih vrijednosti se može upotrijebiti hi kvadrat test. Ovako opisan algoritam se u literaturi zove *PoV* (*Pairs of Values*). Koraci PoV algoritma su sljedeći (uz pretpostavku da je dana slika sivih razina; slika u boji se može promatrati kao slika s tri puta više sivih razina) [6], [9]:

1. Napravi se histogram slike: Za svaku vrijednost sive razine k se odredi broj njenih pojavljivanja u slici n_k.
2. Za svaku parnu sivu razinu $2k$ i odgovarajuću neparnu sivu razinu $2k+1$ se odredi aritmetička sredina:

$$m_k = \frac{n_{2k} + n_{2k+1}}{2} \tag{4.1}$$

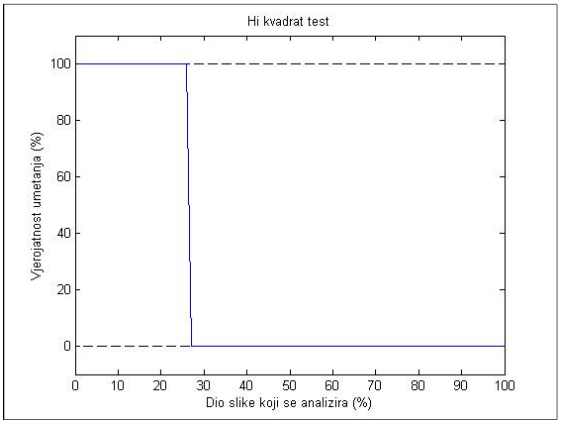
3. Na temelju prethodnih varijabli se odredi vrijednost χ_{N-1}:

$$\chi_{N-1} = \sum_{k=1}^{128} \frac{(m_k - n_{2k})^2}{m_k} \tag{4.2}$$

   Pri tome se izuzimaju pribrojnici za koje je $n_{2k} + n_{2k+1} \leq 4$. Broj N je broj svih pribrojnika uzetih u obzir.
4. Na temelju formule (4.3) se odredi vjerojatnost postojanja skrivene poruke.

$$p = 1 - \int_0^{\chi_{N-1}} \frac{x^{\frac{N-1}{2}-1} e^{-\frac{x}{2}}}{2^{\frac{N-1}{2}} \cdot \Gamma\left(\frac{N-1}{2}\right)} dx \tag{4.3}$$

Navedeni algoritam može detektirati skrivenu poruku samo ako ona zauzima veći dio kapaciteta predviđenog za njeno skrivanje (3 bita po slikovnom elementu). Primjerice, ukoliko se sakrije poruka veličine 25% od prostora predviđenog za njeno smještanje, algoritam neće detektirati njenu prisutnost. Ipak, ukoliko se koristi slijedno smještanje poruke u sliku, moguće je doskočiti problemu. Naime, test se može provoditi tako da se promatra samo dio slike. Tako test provodimo, gledajući sliku slijedno, prvo nad 1% ukupne slike, pa nad 2% ukupne slike, i tako dalje. Na taj način je moguće, ne samo detektirati poruku, već i procijeniti njenu duljinu. Ovdje je uzeto da je poruka u postocima velika kao udio slike za koji je test davao vjerojatnost nešto veću od 0.5. Stvarna veličina se tada preračuna kao dobiveni postotak pomnožen s kapacitetom. Valja napomenuti da za male udjele slike rezultate hi kvadrat testa treba uzimati s dozom opreza jer se test provodi nad relativno malim skupom uzoraka.



Slika 4.7 Graf vjerojatnosti umetanja (sLSB)

Na slici 4.7 je prikazan graf vjerojatnosti umetanja za opisani algoritam. Veličina poruke iznosi 25% ukupnog kapaciteta, a algoritam je procijenio veličinu poruke na 26% ukupnog kapaciteta.

U tablici 4.2 su dani rezultati testiranja PoV algoritma u procjeni duljine skrivene poruke. Pri tome se poruka skrivala slijedno u sliku. Mjere uspješnosti koje se koriste u testiranju metoda

steganalize se mogu pronaći u dodatku A. U skupu od 15 slika u boji se za svaku sliku računalo kolika je apsolutna i relativna razlika procijenjene i stvarne poruke za različite veličine skrivene poruke. U tablici 4.2 su prikazane srednje vrijednosti i standardne devijacije navedenih mjera (standardna devijacija je dana u zagradama). Dakle, u prosjeku se pogriješi za otprilike 5% u odnosu na kapacitet s kojim se raspolaže. U dodatku B se nalazi točan opis svih provedenih ispitivanja.

Tablica 4.2 Uspješnost PoV algoritma u procjeni duljine poruke

Veličina skrivene poruke	25%	50%	75%	**Ukupni prosjek**
Apsolutna razlika (%)	3.2% (5.12%)	6.53% (6.780969%)	5.73% (4.98%)	5.16% (5.74%)
Relativna razlika (%)	12.8% (20.46%)	13.07% (13.56%)	7.65% (6.64%)	11.17% (14.57%)

Uspješnost detekcije skrivene poruke PoV algoritmom se testirala nad 20 slika u boji. U 10 slika je bila slijedno skrivena poruka veličine 10% ukupnog kapaciteta, dok preostale slike nisu mijenjane. Slika je proglašena stego slikom, tj. slikom koja nosi skrivenu informaciju, ukoliko je procijenjena veličina poruke bila veća od 5% ukupnog kapaciteta. Rezultati testiranja su prikazani u tablici 4.3. Rezultati ukazuju na visoku točnost i preciznost PoV metode.

Tablica 4.3 Uspješnost PoV algoritma u detekciji skrivene poruke

Točnost	**Preciznost**	**Odziv**	**Specifičnost**	**F**
0.9	0.83	1.0	0.8	0.91

Veliki nedostatak ove metode je nemogućnost detekcije skrivene poruke kada je ona raspršena po slici, a ne zauzima veći dio kapaciteta predviđenog za skrivanje poruke.

4.4 RS steganaliza

PoV steganaliza promatra samo jednu karakteristiku slike, njen histogram. Pri tome se zanemaruju prostorni rasporedi slikovnih elemenata u slici koji mogu biti korisni u detekciji skrivene poruke. *RS* tehnika steganalize iskorištava prostorne korelacije [6]: time je moguće detektirati skrivenu poruku i procijeniti njenu duljinu čak i kada je poruka raspršena po slici.

Radi jednostavnosti, neka je dana slika sivih razina s vrijednostima sivih razina od 0 do 255 (slika u boji se može promatrati kao tri slike sivih razina). Za grupu G od n slikovnih elemenata ($G = (x_1, x_2, \ldots, x_n)$) se definira diskriminacijska funkcija $f(G)$ koja daje korelaciju među slikovnim elementima grupe [12]:

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} |x_i - x_{i+1}| \tag{4.4}$$

Diskriminacijska funkcija definirana izrazom (4.4) mjeri koliko je grupa slikovnih elemenata *glatka*. Skrivanje poruke LSB metodom smanjuje glatkoću grupe *G*, a time povećava vrijednost funkcije *f*(*G*) [6].

Skrivanje LSB metodom se formalno može opisati funkcijom F_1 koja radi preslikavanja oblika: 0↔1, 2↔3, …, 254↔255. Uz to se definira i dualna funkcija F_{-1} kao: -1↔0, 1↔2, …, 255↔256. Veza između te dvije funkcije je dana izrazom (4.5).

$$F_{-1}(x) = F_1(x+1) - 1 \tag{4.5}$$

Radi potpunosti se definira i funkcija identiteta F_0 dana izrazom (4.6). Funkcije F_1, F_{-1} i F_0 se zovu funkcije prevrtanja [12].

$$F_0(x) = x \tag{4.6}$$

Funkciju prevrtanja se može definirati i na grupi. U tu svrhu se uvodi maska $M = (M_1, M_2, \ldots, M_n)$ koja određuje koja će se funkcija prevrtanja koristiti nad pojedinim elementom grupe *G*. Funkcija prevrtanja F_M se definira kao:

$$F_M(x_1, x_2, \ldots, x_n) = (F_{M_1}(x_1), F_{M_2}(x_2), \ldots, F_{M_n}(x_n)) \tag{4.7}$$

Primjerice, za grupu $G = (x_1, x_2, x_3)$ i masku $M = (1, 0, -1)$ vrijednost funkcije prevrtanja je $F_M(G) = (F_1(x_1), F_0(x_2), F_{-1}(x_3))$. Primjena funkcija prevrtanja češće vodi povećanju vrijednosti diskriminacijske funkcije. Naime, prevrtanjem se povećava zašumljenost slike (više je susjednih slikovnih elemenata koji su različiti). Naravno, to vrijedi kada maska *M* određuje različite funkcije prevrtanja: primjerice, nije nad svakim slikovnim elementom zadano samo prevrtanje F_1.

Ovisno o tome kako funkcija prevrtanja grupe utječe na promjenu vrijednosti diskriminacijske funkcije definiraju se tri tipa grupe [11]:

- Regularna grupa (*regular group*): Grupa *G* je regularna ako vrijedi $f(F_M(G)) > f(G)$.
- Singularna grupa (*singular group*): Grupa *G* je singularna ako vrijedi $f(F_M(G)) < f(G)$.
- Nepromijenjena grupa (*unchanged group*): Grupa *G* je nepromijenjena grupa ako vrijedi $f(F_M(G)) = f(G)$.

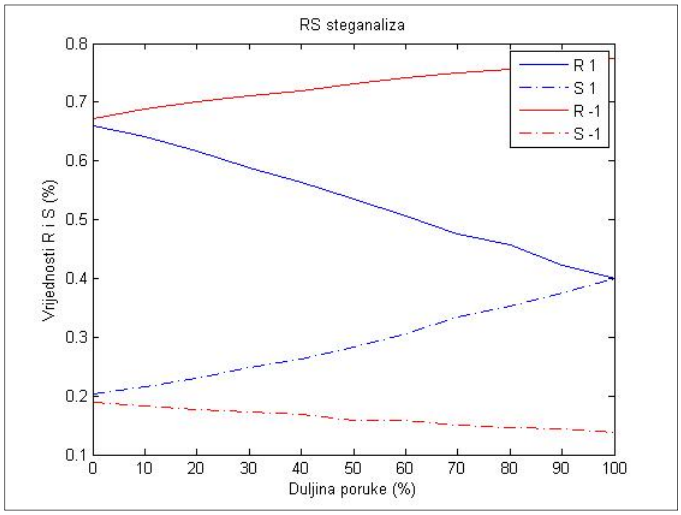
Neka maska *M* sadržava samo vrijednosti 1 i 0 (ne-negativne vrijednosti). Neka je R_M relativni broj regularnih grupa, a S_M relativni broj singularnih grupa nakon primijenjene operacije prevrtanja F_M na cijeloj slici. Budući da su R_M i S_M relativne veličine, vrijedi $R_M + S_M \leq 1$. Analogno se definiraju i oznake R_{-M} i S_{-M} za operaciju prevrtanja F_{-M}. Za R_{-M} i S_{-M} vrijedi $R_{-M} + S_{-M} \leq 1$. U normalnoj slici su valjani sljedeći odnosi [6], [11], [12]:

$$\begin{aligned} R_M &\approx R_{-M} \\ S_M &\approx S_{-M} \\ R_M &> S_M \\ R_{-M} &> S_{-M} \end{aligned} \tag{4.8}$$

Već je prije spomenuto kako funkcija prevrtanja češće vodi povećanju vrijednosti diskriminacijske funkcije. Time se mogu objasniti veće vrijednosti veličina *R* od veličina *S* u normalnoj slici. Jednakost *R* veličina, odnosno *S* veličina, se može pojasniti promatranjem

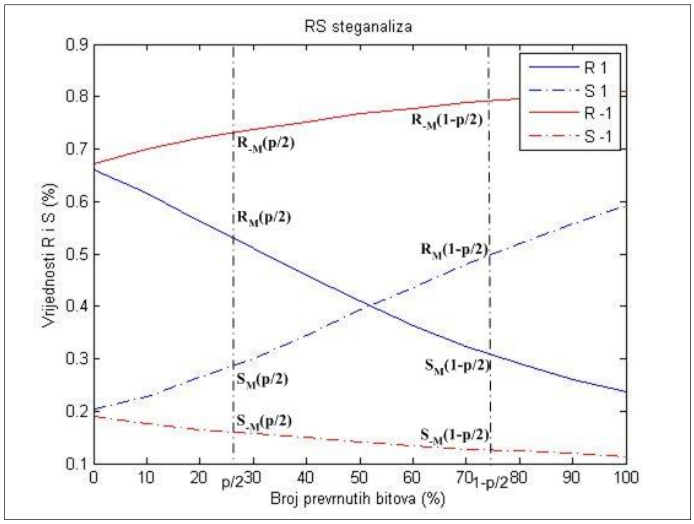
funkcije diskriminacije kao mjere koja određuje koliko je slika glatka. Naime, promatrajući funkcije F_1 i F_{-1} može se uočiti da je veličina R_{-M} za prvotnu sliku jednaka veličini R_M za sliku koja se iz prvotne slike dobiva umanjivanjem slikovnih elemenata za 1. Budući da funkcija f mjeri koliko je slika glatka, relativni broj regularnih i singularnih grupa se ne bi trebao promijeniti kada se vrijednosti slikovnih elemenata normalne slike umanje za 1 [6]. Na slici 4.8 je prikazano kako se mijenjaju veličine R i S s povećanjem duljine poruke koja se skriva. Pri tome je bitno naglasiti da se skriva poruka koja se ravna po uniformnoj razdiobi pa se efektivno promijeni samo pola najmanje značajnih bitova u slici. Kako se povećava duljina skrivene poruke, tako se razlika između vrijednosti R_M i S_M smanjuje i za poruku koja u potpunosti zauzima kapacitet dan za skrivanje postaje jednaka 0. Razlog tomu je povećanje stupnja zašumljenosti slike umetanjem tajne poruke. Za razliku od vrijednosti R_M i S_M, razlika vrijednosti R_{-M} i S_{-M} se povećava kako se povećava duljina poruke koja se skriva. Pojednostavljeno objašnjenje za to se može pronaći u [6].

Uzrok opisanih kretanja R i S veličina u ovisnosti o duljini poruke leži u funkcijama prevrtanja. Neka se u grupi G promatraju slikovni elementi x_i i x_{i+1}. Ako slikovni elementi imaju istu vrijednost, tada djelovanje funkcije prevrtanja F_1 ili F_{-1} nad jednim od njih uzrokuje povećanje njihove apsolutne razlike za jedan. U normalnoj slici je veliki broj parova x_i i x_{i+1} koji imaju iste vrijednosti, što znači da će diskriminacijska funkcija rasti primjenom funkcija prevrtanja na opisani način. Stoga je za očekivati da u normalnoj slici vrijede odnosi (4.8). Ako su slikovni elementi x_i i x_{i+1} LSB susjedi (samo im je najmanje značajan bit različit), tada primjena funkcije prevrtanja F_1 nad oba slikovna elementa ne mijenja njihovu apsolutnu razliku, a apsolutna razlika postaje jednaka nuli kada se ista funkcija prevrtanja upotrijebi samo nad jednim elementom. Kako je u normalnoj slici veliki broj parova x_i i x_{i+1} koji imaju iste vrijednosti, u stego slici broj LSB susjeda x_i i x_{i+1} raste s porastom veličine umetnute poruke. Zbog toga razlika između vrijednosti R_M i S_M pada s porastom veličine poruke koja se skriva. S druge strane, djelovanje funkcije F_{-1} na LSB susjede x_i i x_{i+1} ima suprotan učinak. Naime, primjenom funkcije F_{-1} se povećava apsolutna razlika između x_i i x_{i+1}. Stoga razlika između vrijednosti R_{-M} i S_{-M} raste s porastom veličine poruke koja se skriva.



Slika 4.8 Ovisnost R i S veličina o duljini skrivene poruke

Na slici 4.9 je prikazana ovisnost vrijednosti veličina R i S o broju prevrnutih bitova. U prosjeku će poruka duljine p prevrnuti $p/2$ bitova (p je mjereno u postocima u odnosu na ukupni kapacitet). Valja uočiti da graf na slici 4.8 predstavlja samo pola grafa na slici 4.9.



Slika 4.9 Ovisnost R i S veličina o broju prevrnutih bitova

Modeliranje krivulja u grafu na slici 4.9 predstavlja osnovu za određivanje duljine skrivene poruke. Na temelju eksperimentalnih rezultata iz [6] dolazi se do zaključka da se crvene krivulje mogu modelirati polinomom prvog stupnja, a plave polinomom drugog stupnja. Slika 4.9 pruža bolji uvid u situaciju od slike 4.8 jer prikazuje osam točaka koje se koriste pri procjeni duljine poruke. Te točke su $R_M(p/2)$, $R_M(1-p/2)$, $S_M(p/2)$, $S_M(1-p/2)$, $R_{-M}(p/2)$, $R_{-M}(1-p/2)$, $S_{-M}(p/2)$ i $S_{-M}(1-p/2)$. Ukoliko se izračunaju R i S veličine za sliku u kojoj je skrivena poruka duljine p, dobiju se vrijednosti $R_M(p/2)$, $S_M(p/2)$, $R_{-M}(p/2)$ i $S_{-M}(p/2)$. Ako se prevrnu svi najmanje značajni bitovi stego slike i ponovo izračunaju R i S veličine, dobiju se vrijednosti $R_M(1-p/2)$, $S_M(1-p/2)$, $R_{-M}(1-p/2)$ i $S_{-M}(1-p/2)$. Osim osam izračunatih točaka, za procjenu veličine poruke se koriste i sljedeće pretpostavke [11]:

1 Vrijede jednakosti $R_M(0) = R_{-M}(0)$ i $S_M(0) = S_{-M}(0)$. Ovo znači da se krivulje koje prikazuju R vrijednosti sijeku u $p = 0$. Isto vrijedi i za krivulje koje prikazuju S vrijednosti.
2 Vrijedi jednakost $R_M(0.5) = S_M(0.5)$. Ovo znači da se krivulja koja prikazuje vrijednosti R_M i krivulja koja prikazuje vrijednosti S_M sijeku u $p = 1$.
3 Krivulje koje prikazuju R_M i S_M vrijednosti su parabole, a krivulje koje prikazuju R_{-M} i S_{-M} vrijednosti su pravci.

Ako se s x označi broj prevrnutih bitova, tada se vrijednosti R_M, S_M, R_{-M} i S_{-M} mogu promatrati kao funkcije od x. Kako bi se pojednostavnio izvod, vrši se skaliranje intervala $[p/2, 1-p/2]$ na interval $[0, 1]$. Drugim riječima, varijabla x se transformira u varijablu y izrazom (4.9). Duljina poruke p se određuje iz izraza (4.9) u slučaju kad je x jednak 0.

$$y = \frac{x - \frac{p}{2}}{1 - \frac{p}{2} - \frac{p}{2}} = \frac{x - \frac{p}{2}}{1 - p} \Rightarrow p|_{x=0} = \frac{y}{y - \frac{1}{2}} \tag{4.9}$$

Sad se vrijednosti R_M, S_M, R_{-M} i S_{-M} promatraju kao funkcije od y uz oznake R'_M, S'_M, R'_{-M} i S'_{-M}. Zbog skaliranja, tj. jednadžbe (4.9), vrijedi: $R'_M(0) = R_M(p/2)$, $S'_M(0) = S_M(p/2)$, $R'_{-M}(0) = R_{-M}(p/2)$, $S'_{-M}(0) = S_{-M}(p/2)$, $R'_M(0.5) = R_M(0.5)$, $S'_M(0.5) = S_M(0.5)$, $R'_{-M}(0.5) = R_{-M}(0.5)$, $S'_{-M}(0.5) = S_{-M}(0.5)$, $R'_M(1) = R_M(1-p/2)$, $S'_M(1) = S_M(1-p/2)$, $R'_{-M}(1) = R_{-M}(1-p/2)$ i $S'_{-M}(1) = S_{-M}(1-p/2)$. Treća pretpostavka se može zapisati u matematičkom obliku kao:

$$
\begin{aligned}
R'_M(y) &= \alpha_1 \cdot y^2 + \beta_1 \cdot y + \gamma_1 \\
S'_M(y) &= \alpha_2 \cdot y^2 + \beta_2 \cdot y + \gamma_2 \\
R'_{-M}(y) &= \beta_3 \cdot y + \gamma_3 \\
S'_{-M}(y) &= \beta_4 \cdot y + \gamma_4
\end{aligned}
\tag{4.10}
$$

Uvrštavajući vrijednosti $R'_M(0)$, $S'_M(0)$, $R'_{-M}(0)$, $S'_{-M}(0)$, $R'_M(1)$, $S'_M(1)$, $R'_{-M}(1)$ i $S'_{-M}(1)$ u izraze (4.10) dobiva se:

$$
\begin{aligned}
\gamma_1 &= R'_M(0), & \alpha_1 + \beta_1 &= R'_M(1) - R'_M(0) \\
\gamma_2 &= S'_M(0), & \alpha_2 + \beta_2 &= S'_M(1) - S'_M(0) \\
\gamma_3 &= R'_{-M}(0), & \beta_3 &= R'_{-M}(1) - R'_{-M}(0) \\
\gamma_4 &= S'_{-M}(0), & \beta_4 &= S'_{-M}(1) - S'_{-M}(0)
\end{aligned}
\tag{4.11}
$$

Kombinirajući (4.10) i drugu pretpostavku dobiva se:

$$
\frac{1}{4}\alpha_1 + \frac{1}{2}\beta_1 + \gamma_1 = \frac{1}{4}\alpha_2 + \frac{1}{2}\beta_2 + \gamma_2 \tag{4.12}
$$

Kombinirajući (4.10) i prvu pretpostavku dobiva se:

$$
\begin{aligned}
\alpha_1 \cdot y^2 + \beta_1 \cdot y + \gamma_1 &= \beta_3 \cdot y + \gamma_3 \\
\alpha_2 \cdot y^2 + \beta_2 \cdot y + \gamma_2 &= \beta_4 \cdot y + \gamma_4
\end{aligned}
\tag{4.13}
$$

U izrazu (4.13) *y* je jednak za obje jednadžbe pa se jednadžbe mogu oduzeti čime se dobiva:

$$
(\alpha_1 - \alpha_2) \cdot y^2 + (\beta_1 - \beta_2 - \beta_3 + \beta_4) \cdot y + (\gamma_1 - \gamma_2 - \gamma_3 + \gamma_4) = 0 \tag{4.14}
$$

Iz (4.11) i (4.12) se dobiva:

$$
\begin{aligned}
\alpha_1 - \alpha_2 &= 2 \cdot \left(R'_M(1) - S'_M(1) + R'_M(0) - S'_M(0)\right) \\
\beta_1 - \beta_2 &= -\left(R'_M(1) - S'_M(1)\right) - 3 \cdot \left(R'_M(0) - S'_M(0)\right)
\end{aligned}
\tag{4.15}
$$

Obuhvaćajući izraze (4.11), (4.14) i (4.15) dolazi se do kvadratne jednadžbe:

$$
2 \cdot (A + B) \cdot y^2 + (C - D - B - 3 \cdot A) \cdot y + (A - C) = 0 \tag{4.16}
$$

Koeficijenti *A*, *B*, *C* i *D* se određuju prema sljedećim izrazima:

$$
\begin{aligned}
A &= R'_M(0) - S'_M(0) = R_M(p/2) - S_M(p/2) \\
B &= R'_M(1) - S'_M(1) = R_M(1 - p/2) - S_M(1 - p/2) \\
C &= R'_{-M}(0) - S'_{-M}(0) = R_{-M}(p/2) - S_{-M}(p/2) \\
D &= R'_{-M}(1) - S'_{-M}(1) = R_{-M}(1 - p/2) - S_{-M}(1 - p/2)
\end{aligned}
\tag{4.17}
$$

Relacijom (4.18) se određuje duljina skrivene poruke pri čemu je y korijen kvadratne jednadžbe koji je po apsolutnoj vrijednosti manji. Razlog uzimanja manjeg korijena kvadratne jednadžbe se može vidjeti na slici 4.9. Naime, plava parabola je konveksna, a plava crtkana parabola je konkavna. Te dvije parabole će imati dva sjecišta s odgovarajućim pravcima: jedno za $x = 0$, a drugo za $x > 1$. Kako se traži y za $x = 0$, uzima se manje rješenje kvadratne jednadžbe.

$$p = \frac{y}{y - \frac{1}{2}} \tag{4.18}$$

RS steganalizu se može opisati sljedećim koracima:

1. Odredi se veličina grupe G i maska M koja sadržava samo 0 i 1 (ne-negativne vrijednosti). Primjerice, veličina grupe može biti 4 slikovna elementa, a maska M može biti jednaka $M = (0, 1, 1, 0)$.
2. Sliku se podijeli na nepreklapajuće blokove koji sadržavaju onoliko slikovnih elemenata kolika je veličina grupe G. Blokove slike se transformira u vektor tako da ih se čita po *linearnom cik-cak redoslijedu* (*zig-zag scan order*). Opis čitanja po *cik-cak redoslijedu* je dan u poglavlju o JPEG kompresiji. Svaki transformirani vektor čini zasebnu grupu slikovnih elemenata G.
3. Za svaku grupu G se odredi je li grupa regularna, singularna ili nepromijenjena i to se čini u odnosu na maske M i $-M$. Broj regularnih, singularnih i nepromijenjenih grupa određuju iznose veličina $R_M(p/2)$, $S_M(p/2)$, $R_{-M}(p/2)$ i $S_{-M}(p/2)$.
4. Svakom elementu svake grupe se promijeni vrijednost zadnjeg bita. Za svaku promijenjenu grupu G se odredi je li grupa regularna, singularna ili nepromijenjena i to se čini u odnosu na maske M i $-M$. Broj regularnih, singularnih i nepromijenjenih grupa određuju iznose veličina $R_M(1-p/2)$, $S_M(1-p/2)$, $R_{-M}(1-p/2)$ i $S_{-M}(1-p/2)$.
5. Koristeći izraze (4.16), (4.17) i (4.18) izračuna se veličina skrivene poruke p (u postocima). Duljina poruke u bitovima se dobiva množeći p s ukupnim kapacitetom (3 bita po slikovnom elementu za sliku BMP formata).

U tablici 4.4 su dani rezultati testiranja RS steganalize u procjeni duljine skrivene poruke. Poruka se skrivala raspršeno po slici. Za testiranje se koristilo 15 slika u boji. Veličina grupe je bila 4 slikovna elementa, a koristila se maska $M = (0, 1, 1, 0)$. Iz tablice se može očitati da je pogreška ove metode u određivanju veličine poruke u prosjeku oko 3% u odnosu na kapacitet s kojim se raspolaže. Rezultati iz tablica 4.2 i 4.4 ukazuju na to da RS steganaliza preciznije određuje duljinu skrivene poruke od steganalize PoV algoritmom.

Tablica 4.4 Uspješnost RS steganalize u procjeni duljine poruke

Veličina skrivene poruke	25%	50%	75%	**Ukupni prosjek**
Apsolutna razlika (%)	2.98% (2.08%)	3.33% (2.15%)	3.7% (2.65%)	3.34% (2.27%)
Relativna razlika (%)	11.92% (8.32%)	6.66% (4.29%)	4.94% (3.53%)	7.84% (6.4%)

Uspješnost detekcije skrivene poruke RS steganalizom se testirala na sličan način na koji se testirala uspješnost detekcije skrivene poruke PoV algoritmom. Jedina bitna razlika je u tome što se poruka nije skrivala slijedno, već je bila raspršena po slici. Rezultati testiranja su prikazani u tablici 4.5. Rezultati ukazuju na relativno velik broj FP detekcija (slike koje nisu mijenjane, a detektirane su kao stego slike). Mogući uzrok tomu je slabija kvaliteta slika od onih korištenih pri procjeni duljine poruke.

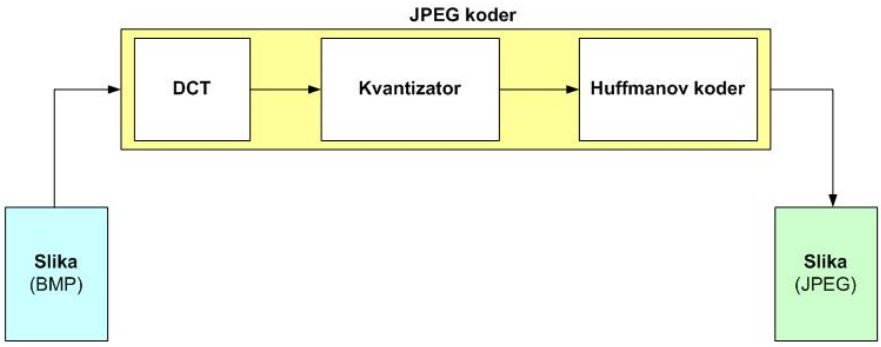
Tablica 4.5 Uspješnost RS steganalize u detekciji skrivene poruke

Točnost	**Preciznost**	**Odziv**	**Specifičnost**	**F**
0.75	0.67	1.0	0.5	0.8

Iz svega navedenog se može zaključiti da se RS steganalizom može s visokom preciznošću procijeniti duljina skrivene poruke čak i kada je poruka raspršena po cijeloj slici.

5. JPEG kompresija

JPEG je jedan od najpopularnijih i najraširenijih formata za zapis slike. Stoga su se razvili steganografski algoritmi prilagođeni upravo tom formatu zapisa. U ovom poglavlju se daje kratak pogled na JPEG kompresiju.



Slika 5.1 JPEG koder

JPEG koder je prikazan na slici 5.1. JPEG koder obavlja sljedeće korake [12], [14], [14] (uz pretpostavku da je ulazna slika u BMP formatu):

1. *Priprema slike*
2. *Transformacija*
3. *Kvantizacija*
4. *Kompresija bez gubitka*

JPEG dekoder obavlja inverzne korake kako bi vratio originalnu sliku. Ipak, treći korak kodiranja uzrokuje gubitke podataka pa nije moguće sliku u potpunosti vratiti u prvotni oblik. U nastavku se ukratko opisuje svaki od navedenih koraka kodiranja.

5.1 Priprema slike

Priprema slike za JPEG koder započinje prebacivanjem slike iz RGB sustava u YCbCr sustav. Y komponenta predstavlja osvijetljenost, a Cb i Cr komponente predstavljaju obojanost slike. Kako je ljudsko oko osjetljivije na osvijetljenost slike nego na obojanost, često se, u cilju smanjenja količine podataka, provodi poduzorkovanje (*subsampling*) Cb i Cr komponenata [14]. Nakon opisanih transformacija svaka komponenta se podijeli na blokove veličine 8×8 slikovnih elemenata i od svakog elementa se oduzme vrijednost 128 (usrednjavanje slikovnih elemenata). U daljnjem procesu se svaka komponenta razmatra kao zasebna slika sivih razina i transformacije koje slijede se obavljaju nad svim komponentama.

5.2 Transformacija

Nakon usrednjavanja slikovnih elemenata, obavlja se 2D DCT transformacija na svakom bloku slike. DCT koeficijenti se računaju prema izrazu (5.1) [14].

$$B(u,v) = \frac{C(u)\cdot C(v)}{4}\sum_{x=0}^{7}\sum_{y=0}^{7} b(x,y)\cdot cos\left(\frac{(2x+1)\pi\cdot u}{16}\right)\cdot cos\left(\frac{(2y+1)\pi\cdot v}{16}\right) \qquad (5.1)$$

U izrazu (5.1) b je izvorni blok u prostornoj domeni, B je transformirani blok, tj. blok u frekvencijskoj domeni, x i y te u i v su indeksi određenog slikovnog elementa u bloku, a vrijednost za C je dana izrazom (5.2).

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}}; u = 0 \\ 1;\ u > 0 \end{cases} \tag{5.2}$$

DCT koeficijent B(0, 0) se naziva i DC koeficijent, a ostali se nazivaju AC koeficijenti. DC koeficijent je jednak srednjoj vrijednosti bloka b. Niže frekvencije (elementi gore lijevo u transformiranom bloku) imaju veće koeficijente od viših frekvencija. Drugim riječima one su dominantne i posjeduju više informacija o slikovnim elementima bloka u prostornoj domeni.

5.3 Kvantizacija

Ljudsko oko dobro uočava razlike u osvijetljenosti površine, ali teško raspoznaje točan doprinos visokih frekvencija u ukupnoj osvijetljenosti [4]. Zbog toga smanjivanje količine informacija u visokim frekvencijama svakog pojedinog bloku ne utječe drastično na percepciju. Smanjivanje se postiže tako da se svaki element transformiranog bloka podijeli odgovarajućim kvantizacijskim faktorom i zaokruži na cjelobrojnu vrijednost. Više frekvencije nakon zaokruživanja poprime vrijednost 0. Time proces uzrokuje gubitak informacije, ali i smanjuje količinu podataka.

Kvantizacijski faktori mogu biti zapisani u kvantizacijskoj matrici veličine 8×8 jer ovise isključivo o poziciji elementa u bloku koji se kvantizira. Formalno se proces kvantizacije može opisati izrazom (5.3).

$$K(u,v) = round\left(\frac{B(u,v)}{Q(u,v)}\right) \tag{5.3}$$

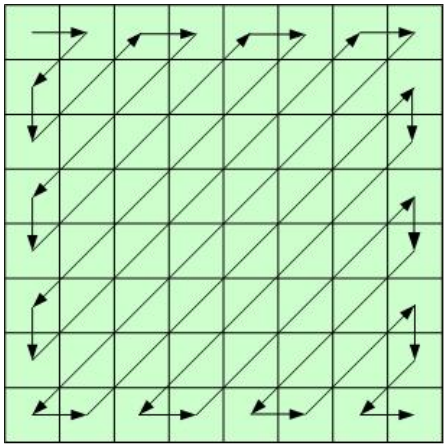
U izrazu (5.3) B je blok koji se kvantizira, K kvantizirani blok, a Q kvantizacijska matrica. Uobičajeno su definirane dvije kvantizacijske matrice, jedna za Y komponentu, a druga za Cb i Cr komponente. Pri tome je kvantizacijska matrica za Cb i Cr komponente podešena tako da uzrokuje veće gubitke nego li matrica za Y komponentu, jer je oko manje osjetljivo na obojanost nego na osvijetljenost. Iako JPEG standard ne definira točne vrijednosti elemenata kvantizacijske matrice, često se koristi *faktor kvalitete* kojim su određene kvantizacijske matrice. Faktor kvalitete poprima vrijednosti od 0 do 100.

5.4 Kompresija bez gubitka

Uobičajeno su susjedni blokovi po srednjoj vrijednosti međusobno slični. Zbog toga se DC koeficijenti diferencijalno kodiraju. To znači da je vrijednost pojedinog DC koeficijenta određena kao vrijednost prethodnog DC koeficijenta uvećanog za zapisanu razliku tih dvaju koeficijenata. Time je potrebno pamtiti samo vrijednost prvog DC koeficijenta i razlike susjednih DC koeficijenata.

Kao što je spomenuto, većina koeficijenata koji se nalaze u donjem desnom dijelu bloka imaju vrijednost 0. Čitanjem po *linearnom cik-cak redoslijedu* (*zig-zag scan order*) se postiže pretvaranje bloka slike u vektor u kojem se većina nula nalazi na kraju vektora. Niz nula na

kraju bloka se zamijeni posebnim znakom za kraj bloka *EOB* (*end of block*) [14]. Čitanje po linearnom *cik-cak* redoslijedu je ilustrirano na slici 5.2.

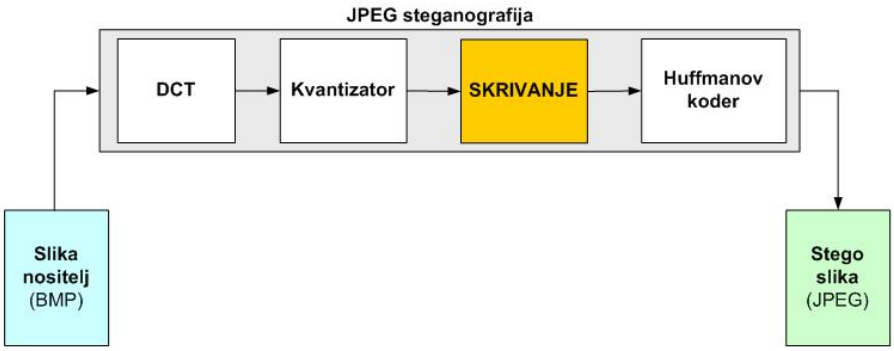


Slika 5.2 Cik-cak redoslijed obilaženja

Na kraju slijedi entropijsko kodiranje. Navedeni zapisi blokova se poredaju jedan iza drugoga, i to od gornjeg lijevog bloka prema donjem desnom bloku, te se vrši kompresija bez gubitka podataka. U tu svrhu se najčešće koristi Huffmanovo kodiranje.

6. Skrivanje u DCT domeni

Steganografske metode koje skrivaju poruku u prostornoj domeni nisu pogodne za rad s formatima slika koje koriste postupak kompresije s gubitkom podataka. Stoga su razvijeni steganografski algoritmi prilagođeni JPEG formatu slika. Specifičnost tih algoritama se ogleda u tome što oni operiraju u DCT domeni pa ih se može svrstati u steganografske tehnike transformacije domene. U principu se za skrivanje koristi JPEG koder pri čemu se između koraka kvantizacije i kompresije bez gubitaka vrši skrivanje tajne poruke. Drugim riječima, poruka se skriva u kvantizirane DCT koeficijente. Na slici 6.1 je prikazan slijed koraka kojeg prate steganografski algoritmi opisani u ovom poglavlju. U postupku dohvaćanja je dovoljno koristiti Huffmanov dekoder kojim se dobivaju DCT koeficijenti u kojima je skrivena poruka.



Slika 6.1 Steganografija prilagođena JPEG standardu

Steganografski algoritmi implementirani u sklopu rada za skrivanje poruke koriste samo komponentu koja definira osvijetljenost slike, tj. Y komponentu. Naime, zbog procesa kodiranja opisanog u prethodnom poglavlju, veći dio kapaciteta predviđenog za skrivanje poruke se nalazi upravo u Y komponenti. Stoga se ne gubi puno na kapacitetu ukoliko se Cb i Cr komponente zanemare pri samom skrivanju.

6.1 Algoritmi zasnovani na LSB supstituciji

Za skrivanje poruka u DCT domeni može se iskoristiti LSB supstitucijska metoda. Najpoznatiji steganografski algoritmi tog tipa su JSTEG i OutGuess.

6.1.1 JSTEG algoritam

Jedan od najjednostavnijih steganografskih algoritama koji radi s JPEG slikama je *JSTEG* algoritam. JSTEG je zapravo LSB metoda koja operira u kvantiziranoj DCT domeni s time da se izbjegavaju DCT koeficijenti čija je vrijednost 0 ili 1. Naime, nule su vezane uz one frekvencije koje se zanemaruju pa bi skrivena poruka mogla biti izgubljena. Jedinice se ne koriste jer u procesu LSB supstitucije mogu poprimiti vrijednost 0. JSTEG algoritam slijedno ide po DCT koeficijentima, tj. ne koristi steganografski ključ. Na slici 6.2 je dan pseudokod JSTEG algoritma implementiranog u okviru rada.

Skrivena poruka se dohvaća algoritmom prikazanim na slici 6.3. Valja primijetiti da se pri dohvaćanju skrivene poruke ne koriste DCT transformacija i kvantizator, već samo Huffmanov dekoder. Na slici 6.4 je prikazan odnos originalne slike (slike nositelja) i stego slike. Skrivena poruka zauzima oko 90% kapaciteta namijenjenog za spremanje tajne poruke

(90% zadnjih bitova DCT koeficijenata koji su različiti od 0 ili 1). Slika nositelj je u BMP formatu, dok je stego slika u JPEG formatu. Zbog toga je stego slika slabije kvalitete od slike nositelja.

```
JSTEG_sakrij(slika, poruka){
    bit_info = bitovi(poruka);
    vel_por = bitovi(duljina(bit_info)); //veličina je spremljena u 32 bita
    stego = Kvantizator(DCT2D(slika));
    Za(i = 0; i < 32; i++){
        DCT_koef = DCT_koeficijent(stego, 'Sljedeći', '!= 0', '!= 1');
        LSB(DCT_koef) = vel_por[i];
    }
    Za(i = 0; i < duljina(bit_info); i++){
        DCT_koef = DCT_koeficijent(stego, 'Sljedeći', '!= 0', '!= 1');
        LSB(DCT_koef) = bit_info[i];
    }
    stego= Huffmanov_koder(stego);
    Vrati stego;
}
```

Slika 6.2 Pseudokod skrivanja poruke JSTEG algoritmom

```
JSTEG_dohvati(stego){
    stego= Huffmanov_dekoder(stego);
    Za(i = 0; i < 32){
        DCT_koef = DCT_koeficijent(stego, 'Sljedeći', '!= 0', '!= 1');
        vel_por[i] = LSB(DCT_koef);
    }
    duljina_poruke = int(vel_por);
    Za(i = 0; i < duljina_poruke; i++){
        DCT_koef = DCT_koeficijent(stego, 'Sljedeći', '!= 0', '!= 1');
        bit_info[i] = LSB(DCT_koef);
    }
    poruka = char(bit_info);
    Vrati poruka;
}
```

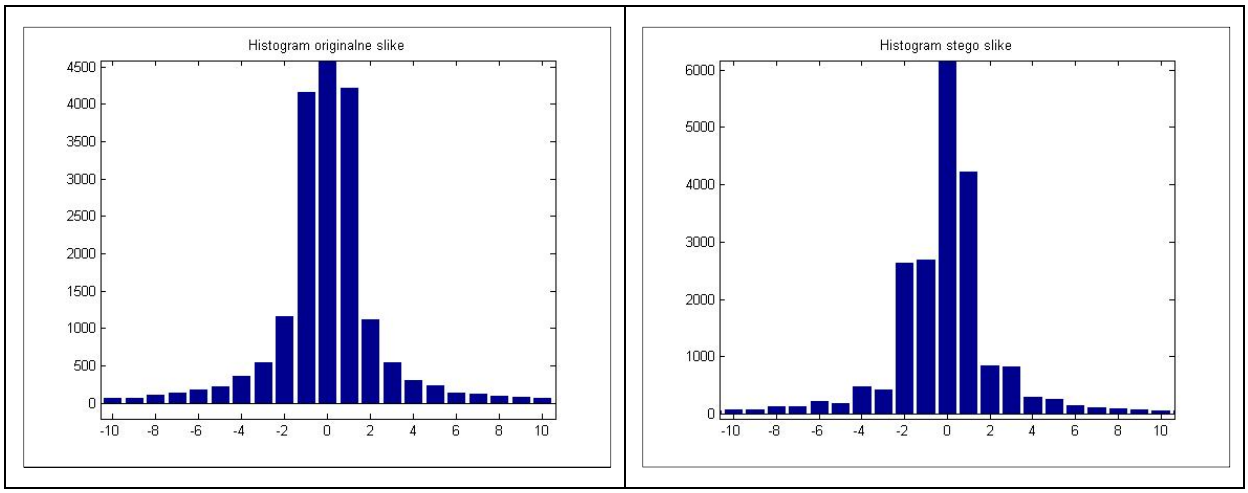
Slika 6.3 Pseudokod dohvaćanja skrivene poruke (JSTEG)

Slika 6.4 Originalna slika i stego slika (JSTEG)

Zanimljivo je pogledati koliki je kapacitet slike u koju se smješta poruka u odnosu na njenu veličinu. Naime, JSTEG za skrivanje ne koristi koeficijent čija je vrijednost 0 ili 1 pa se u usporedbi s jednostavnim LSB algoritmom, opisanim u četvrtom poglavlju, može sakriti manja količina podataka. Test se proveo nad 20 slika pri čemu se promatrala samo Y komponenta YCbCr sustava. Rezultat testiranja pokazuje da je relativni kapacitet 12.34% uz standardnu devijaciju jednaku 4.91%.

6.1.2 Steganaliza hi kvadrat testom

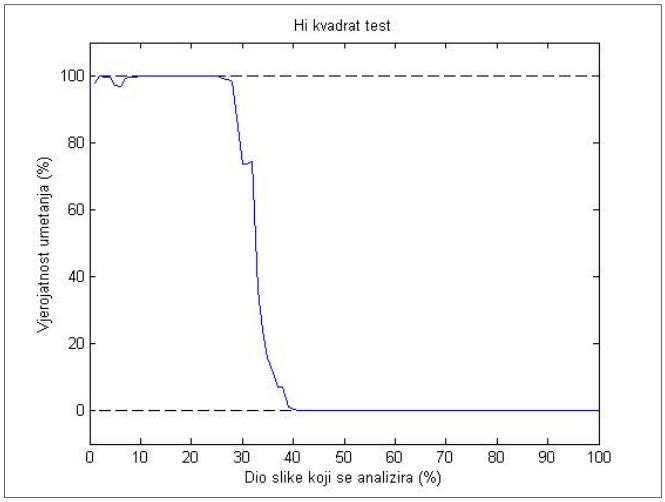
S obzirom da se JSTEG temelji na LSB supstituciji i slijednom umetanju, postavlja se pitanje primjenjivosti hi kvadrat testa u analizi stego slika dobivenih JSTEG algoritmom. Na slici 6.5 su prikazani histogrami DCT koeficijenata originalne slike i stego slike. Radi bolje preglednosti je uzet samo dio histograma i on je uvećan. Zbog toga se, primjerice, ne može iščitati vrijednost histograma za 0. U stego sliku je skrivena poruka veličine 100% kapaciteta. Izuzmu li se vrijednosti histograma za 0 i 1, može se uočiti da su vrijednosti histograma stego slike za parne i susjedne neparne vrijednosti približno jednake. Primjerice, vrijednosti histograma za -1 i -2 ili za 2 i 3 su približno jednake. Uzrok tomu je korištenje LSB umetanja. Također se može uočiti da isto ne vrijedi za histogram originalne slike. Navedena situacija vrijedi i općenito te se može iskoristiti u detekciji skrivene poruke.



Slika 6.5 Histogram DCT koeficijenata originalne slike i stego slike

Kao i u četvrtom poglavlju, i ovdje koristimo hi kvadrat test za detekciju. Algoritam detekcije je gotovo istovjetan onome danom u četvrtom poglavlju: jedino što je potrebno izmijeniti je prvi korak u detekciji. Umjesto histograma slike radi se histogram DCT koeficijenata. Pri tome se izuzimaju koeficijenti čija je vrijednost jednaka 0 ili 1.

Procjenjivanje duljine poruke se vrši na isti način kao u četvrtom poglavlju. Test se provodi promatrajući DCT koeficijente slijedno, prvo se uzima njih 1%, zatim 2%, i tako dalje. Uzima se da je veličina poruka u postocima jednaka relativnom broju DCT koeficijenata za koji je test dao vjerojatnost nešto veću od 0.5. Stvarna veličina se tada preračuna kao dobiveni postotak pomnožen s kapacitetom. Na slici 6.6 je prikazan graf vjerojatnosti umetanja za opisani algoritam. Veličina poruke iznosi 25% ukupnog kapaciteta, a algoritam je procijenio veličinu poruke na 32% ukupnog kapaciteta. Taj rezultat, kao i graf na slici 6.6, daju naznaku da ova metoda nešto slabije procjenjuje veličinu poruke nego njoj ekvivalentna metoda koja operira u prostornoj domeni.



Slika 6.6 Graf vjerojatnosti umetanja (JSTEG)

U tablici 6.1 su dani rezultati testiranja. Testiranje je obavljeno nad 15 slika. Rezultati ukazuju na pogrešku od otprilike 18% u odnosu na kapacitet s kojim se raspolaže. To potvrđuje ranije danu naznaku.

Tablica 6.1 Uspješnost procjene duljine poruke (hi kvadrat test)

Veličina skrivene poruke	25%	50%	75%	**Ukupni prosjek**
Apsolutna razlika (%)	14.10% (3.66%)	18.16% (6.25%)	22.25% (3.97%)	18.17% (5.75%)
Relativna razlika (%)	56.48% (14.66%)	36.34% (12.53%)	29.68% (5.3%)	40.83% (16.12%)

Uspješnost detekcije skrivene poruke je mjerena kao s prethodnim algoritmima steganalize. Pri tome se za skrivanje poruke koristio JSTEG algoritam. Iako se ne može s visokom točnošću procijeniti duljina poruke, rezultati iz tablice 6.2 upućuju na zaključak da predstavljeni algoritam uspješno obavlja detekciju skrivene poruke.

Tablica 6.2 Uspješnost detekcije skrivene poruke (hi kvadrat test)

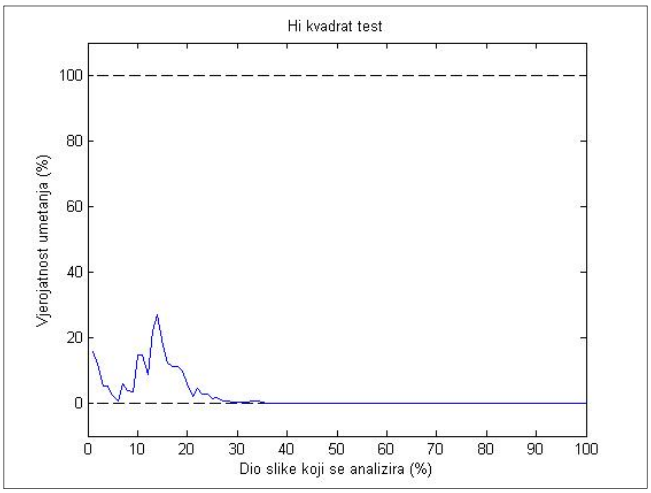
Točnost	**Preciznost**	**Odziv**	**Specifičnost**	**F**
0.9	1.0	0.8	1.0	0.89

6.1.3 Proširenje JSTEG algoritma

Logično proširenje JSTEG algoritma je korištenje generatora pseudo-slučajnih brojeva. Iako se uz to proširenje algoritam može referencirati kao jedna od verzija *OutGuess* algoritma [15], ovdje će oznaka za taj algoritam biti *JSTEG_ext* kako bi se izbjegle zabune sa *OutGuess* algoritmom u odjeljku 6.1.5. Osim što koristi PRNG, JSTEG_ext implementiran u radu koristi i RC4 algoritam za kriptiranje tajne poruke. Kako bi se iz pseudokodova na slikama 6.2 i 6.3 dobio JSTEG_ext potrebno je ulaznu poruku kriptirati RC4 algoritmom koristeći zadani ključ,

inicijalizirati PRNG zadanim ključem te funkciju `DCT_koeficijent(stego, 'Sljedeći', '!= 0', '!= 1')` zamijeniti funkcijom `DCT_koeficijent(stego, 'PRNG', '!= 0', '!= 1')`.

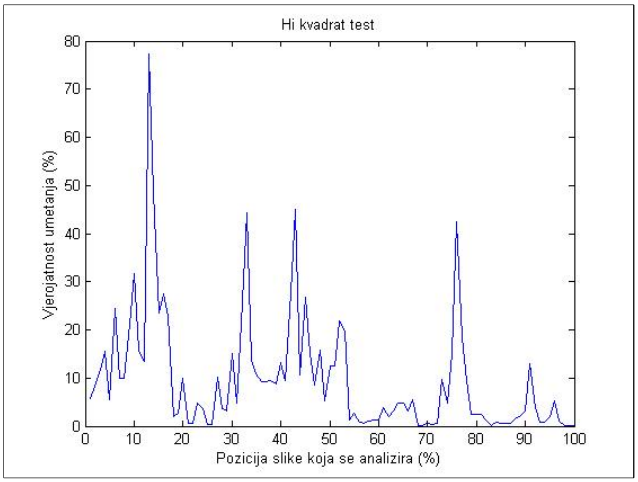
Ukoliko se JSTEG_ext algoritmom skriva poruka koja ne zauzima veći dio kapaciteta, hi kvadrat test ju neće detektirati. Na slici 6.7 je prikazan graf vjerojatnosti umetanja za poruku skrivenu JSEG_algoritmom i tehniku steganalize hi kvadrat testom. Veličina skrivene poruke iznosi 50% ukupnog kapaciteta.



Slika 6.7 Graf vjerojatnosti umetanja (JSTEG_ext)

6.1.4 Steganaliza proširenim hi kvadrat testom

Kako bi se detektirala poruka skrivena DCT_ext algoritmom, potrebno je modificirati hi kvadrat test. Umjesto da se test provodi od jedne pozicije povećavajući broj uzoraka (DCT koeficijenata), prošireni hi kvadrat test uzima uvijek jednak dio slike, ali sa različitih pozicija. Drugim riječima, hi kvadrat test se obavlja lokalno, ali kroz cijelu sliku. Time se za svaki djelić slike nad kojim je obavljen hi kvadrat test dobiva vjerojatnost umetanja. Na slici 6.8 je prikazana vjerojatnost umetanja dobivena hi kvadrat testom provedenim nad različitim pozicijama u slici. U sliku je skrivena poruka veličine 50% kapaciteta, a veličina lokalnog dijela slike koji se testira je jednaka 7.6% kapaciteta.



Slika 6.8 Vjerojatnost umetanja u ovisnosti o poziciji slike

Opisana metoda podsjeća na tehniku kliznog prozora: prozor definira dio slike nad kojim se obavlja hi kvadrat test, a pomičući prozor kroz sliku, test obavljamo nad različitim pozicijama slike. Treba napomenuti da nije svejedno kolika će biti veličina kliznog prozora. Poželjna je mala veličina prozora, ali takva da daje negativne rezultate kada nema skrivene poruke [16]. U tu svrhu se provodi prošireni hi kvadrat test nad slikom, ali se pri testiranju promatraju susjedni DCT koeficijenti koji bi trebali dati negativni rezultat. U tom slučaju se u hi kvadrat testu aritmetička sredina ne računa izrazom (4.1), već izrazom (6.1). Za nalaženje pogodne veličine kliznog prozora se može upotrijebiti binarno pretraživanje kao u [16] [16], no ovdje se radi bolje preciznosti određivanja duljine skrivene poruke obavlja linearno pretraživanje.

$$m_k = \frac{n_{2k} + n_{2k-1}}{2} \tag{6.1}$$

U [15] se detekcija skrivene poruke vrši tako da se zbroje sve izračunate vjerojatnosti i ukoliko je zbroj veći od definiranog praga, slika se proglašava stego slikom. Ipak, slika 6.8 daje naznaku kako bi se osim same detekcije, mogla obaviti i procjena veličine poruke. To se može postići tako da se za svaki dio slike nad kojim je izvršen hi kvadrat test pogleda prelazi li vjerojatnost dobivena testom prethodno definirani prag. Relativni broj dijelova slike čija vjerojatnost prelazi zadani prag određuje veličinu skrivene poruke. Vrijednost praga je u radu postavljena u odnosu na srednju vrijednost izračunatih vjerojatnosti. Testiranje na 5 slika je pokazalo da je pogodno uzeti vrijednost praga jednaku 50% od srednje vrijednosti izračunatih vjerojatnosti. Prilikom testiranja skrivane su poruke veličine 25%, 50% i 75% ukupnog kapaciteta.

Prošireni hi kvadrat test implementiran u sklopu rada prolazi kroz sljedeće korake:

1. Odredi se prikladna veličina kliznog prozora. To se obavlja linearnim pretraživanjem: početna veličina prozora se postavi na 1% slike i povećava se za 0.1% sve dok se hi kvadrat testom ne zadovolji uvjet negativnog rezultata (primjerice dobivene vjerojatnosti su manje od 10%). Pri tome su susjedne vrijednosti u histogramu određene indeksima 2*k* i 2*k*-1. Hi kvadrat test se obavlja koristeći klizni prozor: testira se cijela slika, ali po dijelovima obuhvaćenim kliznim prozorom.
2. Koristeći klizni prozor, kroz sliku se obavlja hi kvadrat test iz odjeljka 6.1.2. Dobivene vjerojatnosti se pamte.
3. Procijenjena veličina poruke je jednaka relativnom broju vjerojatnosti čija je vrijednost veća od praga koji je jednak 50% srednje vrijednosti izračunatih vjerojatnosti.

Procijenjena veličina skrivene poruke iznosi 50.5% kapaciteta za sliku analiziranu na slici 6.8.

Tablica 6.3 Uspješnost procjene duljine poruke (prošireni hi kvadrat test)

Veličina skrivene poruke	25%	50%	75%	**Ukupni prosjek**
Apsolutna razlika (%)	6.2% (4.6%)	7.18% (4.9%)	9.9% (3.76%)	7.76% (4.62%)
Relativna razlika (%)	24.81% (18.45%)	14.37% (9.8%)	13.2% (5.0%)	17.46% (13.22%)

Uspješnost detekcije se određuje kao i ranije uz razliku što je u ovom slučaju korišten JSTEG_ext algoritam. Iz rezultata se može iščitati da se proširenim hi kvadrat test pokazao puno boljim u procjeni veličine poruke od običnog hi kvadrat testa. S druge strane, rezultati prikazani u tablicama 6.2 i 6.4 ukazuju na to da je detekcija proširenim hi kvadrat testom nešto lošija nego detekcija običnim hi kvadrat testom. Pri ovakvoj usporedbi treba imati na umu da JSTEG_ext raspršuje poruku po cijeloj slici, dok JSTEG poruku skriva slijedno.

Tablica 6.4 Uspješnost detekcije skrivene poruke (prošireni hi kvadrat test)

Točnost	**Preciznost**	**Odziv**	**Specifičnost**	**F**
0.85	0.89	0.8	0.9	0.84

6.1.5 OutGuess algoritam

Outguess algoritam, odnosno verzija 0.2 OutGuess algoritma, predstavlja poboljšanje JSTEG_ext algoritma. Algoritam nakon umetanja tajne poruke vrši dodatne modifikacije slike, tj. njenih DCT koeficijenata, kako bi histogram stego slike uskladio s histogramom originalne slike i time spriječio napad proširenim hi kvadrat testom. Na slici 6.9 je dan pseudokod OutGuess algoritma implementiranog u okviru rada.

```
OutGuess_sakrij(slika, poruka, ključ){
    stego = Kvantizator(DCT2D(slika));

    bit_info = bitovi(RC4(poruka, ključ));

    rnd_sjeme = Odredi_prikadno_sjeme(stego, bit_info, 'Iteracija = 5');

    PRNG_sjeme(ključ);
    DCT_koef1 = DCT_koeficijenti(stego, 'PRNG', '!= 0', '!= 1', 'Int:4');

    PRNG_sjeme(rnd_sjeme);
    DCT_koef2 = DCT_koeficijenti(stego, 'PRNG', '!= 0', '!= 1', 'Int:MAX');

    vel_por = bitovi(duljina(bit_info)); //veličina je spremljena u 16 bita
    bit_rs = bitovi(rnd_sjeme); //spremljeno u 16 bita

    Za(i = 0; i < 16; i++){
        LSB(DCT_koef1[i]) = vel_por[i];
    }
    Za(i = 0; i < 16; i++){
        LSB(DCT_koef1[i+16]) = bit_rs[i];
    }
    Za(i = 0; i < duljina(bit_info); i++){
        LSB(DCT_koef2[i]) = bit_info[i];
    }
    stego = Modificiraj(stego);

    stego= Huffmanov_koder(stego);
    Vrati stego;
}
```

Slika 6.9 Pseudokod skrivanja poruke OutGuess algoritmom

U pseudokodu prikazanom na slici 6.9 se može primijetiti nekoliko novina. Prva je funkcija `Odredi_prikadno_sjeme(stego, bit_info, 'Iteracija = 5')`. Ta funkcija nastoji

odabrati ono sjeme generatora slučajnih brojeva za koje će se vršiti najmanje promjene u slici prilikom umetanja tajne poruke. U principu se određivanje obavlja višestrukim izvođenjem dijela algoritma koji vrši umetanje. Broj iteracija se može zadati, a u pseudokodu je taj broj postavljen na 5. Valja primijetiti da se koriste dva sjemena za PRNG: jedno je steganografski ključ (kod smještanja veličine poruke i generiranog sjemena), a drugo je generirano sjeme (kod smještanja šifrirane poruke).

Druga novina je način odabira DCT koeficijenata u koje će se smjestiti poruka. Budući da se poruka prije skrivanja kriptira RC4 algoritmom, bitovi koji se skrivaju imaju oblik slučajne poruke. Stoga nije bitno hoće li se bitovi ubacivati po redu ili ispremiješano. Ono što je bitno je da su efektivno razbacani po cijeloj slici. OutGuess algoritam redom uzima kriptirane bitove i skriva ih obilazeći redom DCT koeficijente. Odabir DCT koeficijenta koji će smjestiti nadolazeći bit kriptirane poruke se odvija u sljedećim koracima:

1. Izračuna se veličina intervala koji pripada tom bitu. To se radi po formuli (6.2). Prikladni su koeficijenti oni koji su različiti od 0 ili 1, a preostali koeficijenti su oni koji se nalaze iza zadnjeg koeficijenta iskorištenog za smještanje.

$$vel\_interval = \frac{2 \cdot broj\_preostalih\_prikladnih\_DCT\_koeficijenata}{broj\_preostalih\_bitova\_poruke} \tag{6.2}$$

2. Generira se slučajan broj *r* iz skupa {1,…, *vel_intervala*}.
3. Odabrani DCT koeficijent je *r*-ti prikladni DCT koeficijent iza zadnjeg koeficijenta iskorištenog za smještanje.

Prvi korak se ne mora obavljati za svaki bit, već ga je dovoljno obaviti za svako nekoliko bitova (primjerice, za svako 8 bitova). Valja primijetiti da se u izrazu (6.2) pojavljuje faktor 2. Razlog tome je što se na svaki bit informacije veže jedan redundantni bit kako bi se u daljnjim koracima mogla uspješno obaviti modifikacija stego slike. Time kapacitet postaje duplo manji od JSTEG algoritma. Također, valja primijetiti da je funkcija koja radi odabir DCT koeficijenata pozvana 2 puta s različitim argumentima. Prvi poziv određuje DCT koeficijente pri smještanju veličine poruke i generiranog sjemena za PRNG. U tom slučaju je veličina intervala fiksna i iznosi 4 bita. Fiksna veličina intervala se koristi kako bi se poruka mogla izvaditi poznavanjem samo steganografskog ključa. Drugi poziv određuje DCT koeficijente kod skrivanja kriptirane poruke i u tom slučaju se veličina intervala računa prema formuli (6.2). Drugi poziv se nastavlja na prvi, tj. uzimaju se u obzir samo oni DCT koeficijenti koji nisu iskorišteni u prvom pozivu funkcije.

Treća novina se ogleda u tome što se osim veličine poruke skriva i generirano sjeme za PRNG. To je nužno napraviti kako bi se mogla izvaditi skrivena poruke.

Zadnja i najvažnija novina je modifikacija stego slike. Algoritam modifikacije je prikazan na slici 6.10. Pseudokod je napravljen na temelju odgovarajućeg pseudokoda u [16] i prati implementaciju ostvarenu u radu. Na početku algoritma se određuju DCT koeficijenti prikladni za smještanje poruke (varijabla `DCT`), njihov broj (varijabla `k`) te histogram stego slike (varijabla `N`). Uz to se računa i skalirajući faktor (varijabla `a`) pomoću kojeg se definiraju pragovi maksimalne greške. Nakon toga se inicijaliziraju brojač pogreške (varijabla `N_error`) i pragovi maksimalne greške (varijabla `N_prag`). Te dvije varijable određuju hoće li se modificirati stego slika u pojedinoj iteraciji: ako brojač pogreške prijeđe odgovarajući

prag slika se nastoji modificirati kako bi se očuvala statistička svojstva promatranog koeficijenta [16]. Nakon inicijalizacije varijabli, slijedi modificiranje slike.

```
Modificiraj(stego){
    DCT = DCT_koeficijenti(stego, 'Slijedno', '!= 0', '!= 1');
    N = dct_histogram(stego);
    k = Broj_dct_koef(stego, '!=0', '!=1');
    a = 0.03*5000/k;
    Za(dct = min(DCT); dct <= max(DCT); dct++){
        N_error[dct] = 0;
        N_prag[dct] = a*N[dct];
    }
    Za(i = 1; i < k; i++){
        Ako(DCT[i] je mijenjan){
            dct_susjed = DCT[i] XOR 1;
            Ako(N_error[dct_susjed] > 0) N_error[dct_susjed]--;
            Inače_ako(N_error[DCT[i]] < N_prag[DCT[i]]) N_error[DCT[i]]++;
            Inače_ako(Promijena(i, DCT[i], DCT) == 0) N_error[DCT[i]]++;
        }
    }
    Za(dct = min(DCT); dct <= max(DCT); dct++){
        Dok(N_error[dct] != 0){
            N_error[dct]--;
            Promijena(k, dct, DCT);
        }
    }
    Vrati stego;
}

Promjena(i, dct, DCT){
    Za(j = i - 1; j >= 0; j--){
        Ako(DCT[j] == dct && . . .
            LSB(DCT[j]) nije bit poruke && . . .
            DCT[j] nije korišten za korekcije){
            DCT[j] = dct XOR 1;
            Vrati 1;
        }
    }
    Vrati 0;
}
```

Slika 6.10 Pseudokod modificiranja stego slike

Za svaki koeficijent različit od 0 ili 1 se provjerava je li on mijenjan tijekom skrivanja slike. Ukoliko nije mijenjan nastavlja se sa sljedećim koeficijentom, a inače je potrebno ažurirati varijable ili modificirati stego sliku. Dakle, za promijenjeni DCT koeficijent se prvo gleda postoji li njegov LSB susjed (koeficijent jednak promatranom osim u najmanje značajnom bitu) koji također sadrži bit poruke i nije uparen s nekim drugim DCT koeficijentom. Ukoliko postoji tada se ta dva koeficijenta uparuju, tj. smanjuje se brojač pogreške za susjedne koeficijente. Ukoliko ne postoji, onda odnos brojača pogreške i praga maksimalne greške određuje hoće li se stego slika modificirati. Ukoliko brojač ne prelazi prag, on se samo uveća za 1. Inače se nastoji modificirati stego slika funkcijom `Promjena`. Ta funkcija traži DCT koeficijent koji ima istu vrijednost kao i promatrani koeficijent, a nije mijenjan, ne sadrži bit poruke i u slici se nalazi se prije promatranog koeficijenta. Ukoliko pronađe takav koeficijent, funkcija mu mijenja najmanje značajan bit i vrati kod uspjeha. Time je neutralizirana promjena u histogramu nastala smještanjem bita poruke u promatrani koeficijent. Ukoliko

funkcija ne pronađe takav koeficijent, brojač pogreške se uveća za 1. Nakon što se obiđe svaki DCT koeficijent različit od 0 ili 1, provodi se petlja u kojoj se pokušava za svaki koeficijent koji sadrži poruku, a nije statistički neutraliziran prethodnim postupkom, pronaći koeficijent koji ima istu vrijednost, a nije mijenjan niti sadrži bit poruke. Ovo se sada obavlja nad cijelom stego sliko. Ukoliko se uspije pronaći takav koeficijent, mijenja mu se najmanje značajan bit. Opisani algoritam nastoji ispuniti sljedeće zahtjeve [16]:

- Za bilo koji dio slike distribucija DCT koeficijenata stego slike je slična distribuciji originalne slike.
- Broj obavljenih korekcija je malen.

Na slici 6.11 je prikazan pseudokod dohvaćanja skrivene poruke. Zanimljivo je pogledati kakva je razlika histograma originalne slike i stego slike te kakav odziv daje prošireni hi kvadrat test za stego sliku dobivenu OutGuess algoritmom.

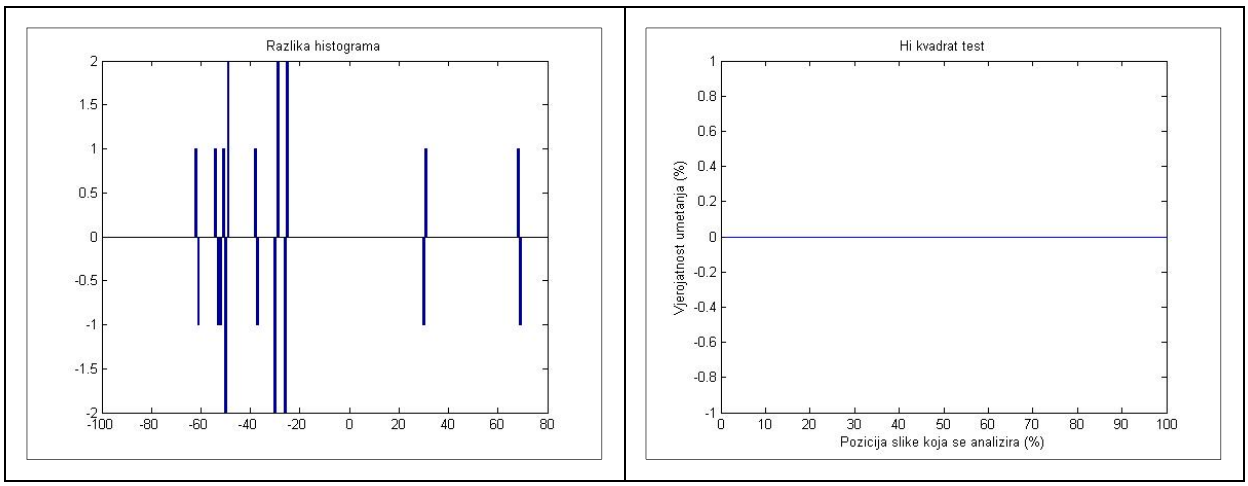
```
OutGuess_dohvati(stego, ključ){
    stego= Huffmanov_dekoder(stego);

    PRNG_sjeme(ključ);
    DCT_koef1 = DCT_koeficijenti(stego, 'PRNG', '!= 0', '!= 1', 'Int:4');

    Za(i = 0; i < 16; i++){
        vel_por[i] = LSB(DCT_koef1[i]);
    }
    Za(i = 0; i < 16; i++){
        bit_rs[i] = LSB(DCT_koef1[i+16]);
    }

    PRNG_sjeme(int(bit_rs));
    DCT_koef2 = DCT_koeficijenti(stego, 'PRNG', '!= 0', '!= 1', 'Int:MAX');

    duljina_poruke = int(vel_por);
    Za(i = 0; i < duljina_poruke; i++){
        bit_info[i] = LSB(DCT_koef2[i]);
    }

    poruka = RC4(char(bit_info), ključ);
    Vrati poruka;
}
```

Slika 6.11 Pseudokod dohvaćanja skrivene poruke (OutGuess)

Na slici 6.12 se može vidjeti razlika histograma originalne i stego slike kao i rezultat proširenog hi kvadrat testa. Veličina tajne poruke koja se skriva iznosi 90% kapaciteta namijenjenog za spremanje tajne poruke. Grafovi pokazuju da su histogrami originalne slike i stego slike dosta slični te da hi kvadrat test ne detektira nikakvu skrivenu poruku. Na slici 6.13 je prikazan odnos originalne slike (slike nositelja) i stego slike. Kao i kod JSTEG algoritma, provedeno je testiranje količine podataka koju se može sakriti OutGuess algoritmom u zadani objekt nositelj. Rezultat testiranja pokazuje da je relativni kapacitet 6.17% uz standardnu devijaciju jednaku 2.45%. To je upravo pola od kapaciteta JSTEG algoritma.

Treba spomenuti da OutGuess algoritam opisan u ovom odjeljku i implementiran u okviru rada ne prati u potpunosti stvarnu specifikaciju OutGuessa. Zanimljiv mehanizam je opisan u [16], a zove se *uvjerljivo poricanje* (*plausible deniability*). Taj mehanizam je koristan kada treća strana može detektirati skrivenu poruku, ali ne može točno kazati koji dijelovi slike su

promijenjeni i koliko je skrivena poruka velika. U tom slučaju je korisno u stego objekt sakriti dvije poruke: jednu tajna, a drugu bezazlenu koja će biti pokazana trećoj strani ukoliko je to neizbježno. Na zahtjev treće strane se provodi dohvaćanje lažne poruke uz poricanje postojanja ikakve druge poruke. Time je izbjegnuto otkrivanje tajne poruke, ali je smanjen kapacitet za njeno skrivanje.



Slika 6.12 Razlika histograma originalne slike i stego slike, prošireni hi kvadrat test

Slika 6.13 Originalna slika i stego slika (OutGuess)

6.1.6 Napad na OutGuess algoritam

Iako OutGuess algoritam radi neznatne modifikacije histograma slike, promjene nastale na slici su dovoljne velike za detektiranje poruke i procjenu njene duljine. Promjene koje OutGuess algoritam radi nad kvantiziranim DCT koeficijentima se odražavaju kroz diskontinuitete u prostornoj domeni na rubovima blokova veličine 8×8 [17]. Dakle, objekt nositelj bi trebao imati manje diskontinuitete nego stego slika. Ovo predstavlja temelj postupka kojim se procjenjuje veličina poruke.

Kako bi se uskladile statistički vidljive distorzije histograma, potrebno je modificirati sliku nakon procesa skrivanja poruke. Kao što je pokazano, bez modifikacije stego slike,

odgovarajuće susjedne vrijednosti u histogramu stego slike se manje razlikuju nego u histogramu originalne slike. Neka se u sliku ubacuje poruka (slučajan niz bitova) veličine α ukupnog kapaciteta. Tada za dvije susjedne vrijednosti u histogramu slike vrijedi [16]:

$$\begin{aligned} h_{2i}{}^{*} &= h_{2i} - \frac{\alpha}{2}(h_{2i} - h_{2i+1}) \\ h_{2i+1}{}^{*} &= h_{2i+1} - \frac{\alpha}{2}(h_{2i+1} - h_{2i}) \end{aligned} \tag{6.3}$$

U izrazu (6.3) h_{2i} i h_{2i+1} su vrijednosti histograma u originalnoj slici, a $h_{2i}{}^{*}$ i $h_{2i+1}{}^{*}$ su vrijednosti histograma u stego slici (i je različit od 0). Faktor α je podijeljen s 2 jer se u prosjeku samo pola bitova promijeni. Izraz (6.3) je posljedica korištenja LSB supstitucijske metode. Ukoliko se želi postići da nakon modifikacije stego slika ima isti histogram kao i originalna slika, potrebno je osigurati da nakon umetanja za svaku moguću vrijednost DCT koeficijenta ostane dovoljno DCT koeficijenata te vrijednosti koji nisu nositelji poruke. Drugim riječima, uz pretpostavku da je h_{2i} veće od h_{2i+1}, mora vrijediti relacija (6.4).

$$(1 - \alpha) \cdot h_{2i+1} \geq \frac{\alpha}{2}(h_{2i} - h_{2i+1}) \tag{6.4}$$

Relaciju (6.4) se može izraziti i u obliku:

$$\alpha \leq \frac{2 \cdot h_{2i+1}}{h_{2i} + h_{2i+1}} \tag{6.5}$$

Relacija (6.5) se može iskoristiti za odabir slike nositelja prilikom smještanja određene poruke, odnosno pri određivanju kapaciteta slike nositelja [16], [17]. Ako se sa α_i označi vrijednost desne strane, tada se može izračunati maksimalna veličina poruke za koju se uspješno obavljaju korekcije izrazom $\alpha = min(\alpha_i)$. Radi jednostavnijeg zapisa u kasnijim razmatranjima, ovdje se uvodi oznaka $a = min(\alpha_i) / 2$. Broj a je zapravo polovina omjera kapaciteta i broja svih DCT koeficijenata. Neka je P broj svih DCT koeficijenata različitih od 0 i 1, a p relativna veličina poruke (veličina poruke u odnosu na kapacitet). Ukupna veličina poruke u bitovima je jednaka $2 \cdot p \cdot a \cdot P$. Očekivani broj promjena za DCT koeficijente vrijednosti $2i$ i $2i+1$ nastao skrivanjem poruke i korekcijom slike iznosi $max(p \cdot a \cdot h_{2i}, p \cdot a \cdot h_{2i+1})$. Ukupni broj promjena nastao skrivanjem poruke i korekcijama stego slike iznosi [17]:

$$T_p = 2pa \sum_{i \neq 0} \dot{H}_{2i} = paP + pa \sum_{i \neq 0} \left| \dot{H}_{2i} - \dot{h}_{2i} \right| \tag{6.6}$$

Varijable $\dot{H}_{2i}$ i $\dot{h}_{2i}$ u izrazu (6.6) su jednake $\dot{H}_{2i} = max(h_{2i}, h_{2i+1})$ i $\dot{h}_{2i} = min(h_{2i}, h_{2i+1})$. Pribrojnik $p \cdot a \cdot P$ je broj promjena nastalih skrivanjem poruke, dok je drugi pribrojnik broj promjena nastalih korekcijom stego slike.

Ako se nakon skrivanja poruke relativne veličine p, sakrije poruka veličine q, očekivani broj promjena za DCT koeficijente vrijednosti $2i$ i $2i+1$ je dan izrazom (6.7). Prije nego se nastavi s analizom, pogledajmo kratak primjer koji će razjasniti genezu izraza (6.7). Neka se u skup od n cijelih brojeva provodi dva puta LSB supstitucijska metoda slučajno birajući podskup brojeva koji će biti mijenjani. Neka je prvi put odabran podskup A od a cijelih brojeva, a drugi put podskup B od b cijelih brojeva. Broj brojeva koji su mijenjani dva puta je jednak $|A \cap B|$. Očekivana veličina skupa $A \cap B$ je jednaka $a \cdot b / n$. Brojevi koji su dva puta mijenjani su na

kraju procesa isti kao i na početku. Drugim riječima, broj brojeva koji nakon dvije provedene LSB supstitucije imaju različite vrijednosti nego na početku procesa je jednak $|A| + |B| - 2\cdot|A\cap B| = a + b - 2\cdot a\cdot b/n$. Sada se, uz pretpostavku da je h_{2i} veće od h_{2i+1}, mogu objasniti relacije (6.7). Za prvu relaciju a je jednak $p\cdot a\cdot\dot{H}_{2i}$ (očekivani broj korekcija za DCT koeficijente vrijednosti 2i uz poruku relativne veličine p), b je jednak $q\cdot a\cdot\dot{H}_{2i}$ (očekivani broj korekcija za DCT koeficijente vrijednosti 2i uz poruku relativne veličine q), a n je jednak $\dot{H}_{2i}$ (broj DCT koeficijente vrijednosti 2i). Analogno se pokazuje za drugu relaciju.

$$\begin{aligned}\Delta h_{2i} &= pa\dot{H}_{2i} + qa\dot{H}_{2i} - 2pqa^2\frac{pa\dot{H}_{2i}\cdot qa\dot{H}_{2i}}{\dot{H}_{2i}} = a\cdot(p+q-2pqa)\cdot\dot{H}_{2i}\\ \Delta h_{2i+1} &= pa\dot{H}_{2i} + qa\dot{H}_{2i} - 2pqa^2\frac{\dot{H}_{2i}^{\,2}}{\dot{h}_{2i}} = a\cdot\left(p+q-2pqa\frac{\dot{H}_{2i}}{\dot{h}_{2i}}\right)\cdot\dot{H}_{2i}\end{aligned} \tag{6.7}$$

Koristeći relacije (6.7) dobiva se ukupan broj promjena pri skrivanju poruka veličine $2\cdot p\cdot a\cdot P$ i $2\cdot q\cdot a\cdot P$. Broj promjena se dobiva zbrajajući vrijednosti Δh_i i iznosi:

$$T_{pq} = 2a\sum_{i\neq 0}\left(p+q-apq\left(1+\frac{\dot{H}_{2i}}{\dot{h}_{2i}}\right)\right)\cdot\dot{H}_{2i} \tag{6.8}$$

Kao što je spomenuto na početku odjeljka, stego slika ima veće diskontinuitete u prostornoj domeni na rubovima blokova veličine 8×8 od slike nositelja. Za sliku I sivih razina rezolucije M×N diskontinuiteti se mjere formulom za *bločnost* (*blockiness*) [15], [17]:

$$B = \sum_{i=1}^{\left\lfloor\frac{M-1}{8}\right\rfloor}\sum_{j=1}^{N}\left|I_{8i,j} - I_{8i+1,j}\right| + \sum_{i=1}^{M}\sum_{j=1}^{\left\lfloor\frac{N-1}{8}\right\rfloor}\left|I_{i,8j} - I_{i,8j+1}\right| \tag{6.9}$$

Treba napomenuti da se bločnost mjeri nad slikom koja je transformirana u prostornu domenu. Eksperimenti provedeni u [17] pokazuju da bločnost *linearno* raste s brojem prevrnutih LSB bitova DCT koeficijenata, odnosno s porastom veličine poruke koja se skriva. Ukoliko se bločnost promatra kao funkciju duljine poruke, tada se bločnost može zapisati u obliku:

$$B(p) = C\cdot T_p + D \tag{6.10}$$

Prva derivacija funkcije $B(p)$, tj. nagib bločnosti, je najveća za sliku nositelja i smanjuje se za stego slike [15]. Ovdje se za prvu derivaciju od $B(p)$ koristi oznaka S (zbog linearnosti od B, S nije ovisan o p).

Veličine bitne za opis algoritma označene su na sljedeći način:

- Prva derivacija od $B(p)$ je označena s S.
- Bločnost slike nositelja u ovisnosti o veličini poruke koja se u nju skriva je označena s $B_0(p)$. Pripadna prva derivacija je označena s S_0.
- Bločnost stego slike u ovisnosti o veličini poruke koja se skriva u stego sliku je označena s $B_s(p)$. Pripadna prva derivacija je označena s S_s.

- Neka je dana slika koja se dobiva skrivanjem poruke relativne veličine 1 u sliku nositelja. Bločnost te slike u ovisnosti o veličini poruke koja se u nju skriva je označena s $B_1(p)$. Pripadna prva derivacija je označena s S_1. Valja primijetiti da je $B_1(0) = B_0(1)$.

Neka je u stego sliku skrivena poruka relativne veličine p. Tada se, koristeći izraze (6.10), (6.8), (6.6) i činjenicu da je S derivacija od B, dobivaju relacije:

$$S_0 = B_0(1) - B_0(0) = C \cdot (T_1 - T_0) = 2aC \sum_{i \neq 0} \dot{H}_{2i}$$

$$S_s = B_s(1) - B_s(0) = C \cdot (T_{p1} - T_{p0}) = 2aC \sum_{i \neq 0} \left(1 - ap\left(1 + \frac{\dot{H}_{2i}}{\dot{h}_{2i}}\right)\right) \cdot \dot{H}_{2i} \tag{6.11}$$

$$S_1 = B_1(1) - B_0(1) = C \cdot (T_{11} - T_{10}) = 2aC \sum_{i \neq 0} \left(1 - a\left(1 + \frac{\dot{H}_{2i}}{\dot{h}_{2i}}\right)\right) \cdot \dot{H}_{2i}$$

Oduzimanjem prve i druge jednadžbe te prve i treće jednadžbe dobivaju se jednakosti:

$$\begin{aligned} S_0 - S_s &= 2pa^2C \sum_{i \neq 0} \left(1 + \frac{\dot{H}_{2i}}{\dot{h}_{2i}}\right) \cdot \dot{H}_{2i} \\ S_0 - S_1 &= 2a^2C \sum_{i \neq 0} \left(1 + \frac{\dot{H}_{2i}}{\dot{h}_{2i}}\right) \cdot \dot{H}_{2i} \end{aligned} \tag{6.12}$$

Sada se dijeleći jednadžbe iz (6.12) određuje relativna duljina poruke p:

$$p = \frac{S_0 - S_s}{S_0 - S_1} \tag{6.13}$$

Dakle, poznavanjem S_0, S_s, S_1, moguće je procijeniti veličinu skrivene poruke. Vrijednost S_s, se računa iz stego slike temeljem $B_s(0)$, $B_s(1)$. Ukoliko nije poznata slika nositelj, nisu poznate ni vrijednosti $B_0(0)$, $B_0(1)$, $B_1(0)$ i $B_1(1)$, pa na prvi pogled relacija (6.13) nije od velike koristi. Ipak, moguće je izvršiti procjenu nepoznatih vrijednosti. To se čini konstrukcijom slike koja će biti srodna originalnoj slici. Konstrukcija se odvija u sljedećim koracima [15], [17]:

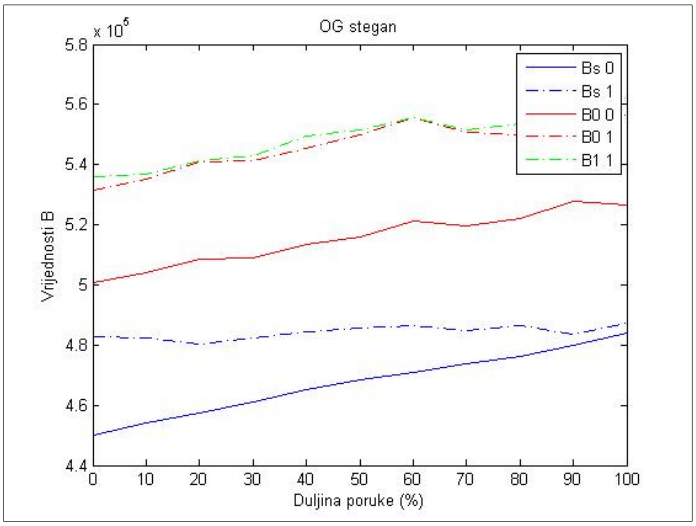
1. Stego slika se prebaci u prostornu domenu.
2. Odbace se 4 stupca stego slike.
3. Na tako modificiranoj slici se izvrši JPEG kompresija koristeći istu kvantizacijsku matricu koja je korištena za stego sliku.

Uzimajući u obzir sve navedeno, koraci detekcije i procjene duljine skrivene poruke su sljedeći [15], [17]:

1. Stego slika se prebaci u prostornu domenu te joj se izračuna bločnost $B_s(0)$.
2. U stego sliku se OutGuess algoritmom sakrije najveća moguća poruka. Dobivena slika se prebaci u prostornu domenu te joj se izračuna bločnost $B_s(1)$.
3. Na temelju $B_s(0)$ i $B_s(1)$ se odredi vrijednost $S_s = B_s(1) - B_s(0)$.

4. Konstruira se slika koja je srodna originalnoj slici prema ranije opisanim koracima. Slika se prebaci u prostornu domenu, te joj se izračuna bločnost $B_0(0)$.
5. U srodnu sliku se OutGuess algoritmom sakrije najveća moguća poruka. Dobivena slika se prebaci u prostoru domenu te joj se izračuna bločnost $B_0(1)$.
6. Na temelju $B_0(0)$ i $B_0(1)$ se odredi vrijednost $S_0 = B_0(1) - B_0(0)$.
7. U sliku dobivena OutGuess algoritmom u petom koraku se sakrije najveća moguća poruka. Dobivena slika se prebaci u prostoru domenu te joj se izračuna bločnost $B_1(1)$.
8. Na temelju $B_0(1)$ i $B_1(1)$ se odredi vrijednost $S_1 = B_1(1) - B_0(1)$.
9. Korištenjem izraza (6.13) se odredi duljina skrivene poruke.

Opisani slijed koraka će se u radu referencirati kao *OG_stegan* algoritam.



Slika 6.14 Kretanje bločnosti u ovisnosti o duljini skrivene poruke

Na slici 6.14 je prikazano kretanje bločnosti u ovisnosti o duljini poruke koja se skriva. Zanimljivo je pogledati kretanje veličine S_s s povećanjem veličine skrivene poruke. Veličina S_s je zapravo jednaka vertikalnom razmaku između plave crtkane crte i plave pune crte. Razmak je maksimalan kada nema skrivene poruke, a minimalan kada se sakrije najveća moguća poruka. Osim toga, za kretanje veličine $B_s(0)$ se može kazati da je pravocrtno, tj. da se $B_s(0)$ linearno povećava s porastom veličine skrivene poruke.

Tablica 6.5 Uspješnost procjene duljine poruke (OG_stegan)

Veličina skrivene poruke	25%	50%	75%	**Ukupni prosjek**
Apsolutna razlika (%)	4.74% (4.5%)	6.93% (5.11%)	7.2% (6.0%)	6.29% (5.22%)
Relativna razlika (%)	19.08% (18.04%)	13.89% (10.25%)	9.6% (7.93%)	14.19% (13.13%)

Mjerenje uspješnosti detekcije je provedeno kao s prethodnim algoritmima uz razliku što je u ovom slučaju za skrivanje korišten OutGuess algoritam. Rezultati pokazuju da je OG_stegan u procjeni duljine poruke dao relativno dobre rezultate u odnosu na obični hi kvadrat test. Nešto slabiji rezultat je dobiven prilikom testiranja detekcije skrivene poruke.

Tablica 6.6 Uspješnost detekcije skrivene poruke (OG_stegan)

Točnost	Preciznost	Odziv	Specifičnost	F
0.85	0.89	0.8	0.9	0.84

Uspješnost procjene veličine skrivene poruke uvelike ovisi o procjeni bločnosti originalne slike. Velike pogreške u procjeni se mogu dogoditi u slučaju duple kompresije. Naime, slika nositelj može biti spremljena u JPEG formatu s kvantizacijskom matricom drugačijom od one koja je korištena za stego sliku. U tom slučaju je potrebno napraviti procjenu kvantizacijske matrice slike nositelja i modificirati proces konstrukcije slike slične originalu. Procedura je detaljno opisana u [17].

6.2 F skupina algoritama

F skupina steganografskih algoritama nije zasnovana na LSB supstitucijskoj metodi, već se umetanje vrši mijenjanjem apsolutne vrijednosti DCT koeficijenata. Ovoj skupini pripadaju *F3*, *F4* i *F5* algoritmi.

6.2.1 F3 algoritam

F3 algoritam predstavlja modifikaciju JSTEG algoritma. Dvije su bitne razlike [18]:

1. Kada najmanje značajan bit DCT koeficijenta nije jednak bitu poruke koji se skriva u taj koeficijent, DCT koeficijentu se ne mijenja najmanje značajan bit, već mu se smanjuje apsolutna vrijednost. Kod skrivanja se koriste i DCT koeficijenti čija je vrijednost jednaka 1, ali se izuzimaju DCT koeficijenti čija je vrijednost 0.
2. Pri skrivanju valja uzeti u obzir proces *sažimanja* (*shrinkage*) koji se događa ubacivanjem bita 0 u DCT koeficijent čija je vrijednost -1 ili 1. Tada je nova vrijednost DCT koeficijenta jednaka 0. Budući da primatelj ne može razlikovati 0 nastalu sažimanjem od 0 koja ne sadrži bitove poruke, potrebno je bit poruke ponovo postaviti u sljedeći DCT koeficijent.

Skrivanje poruke F3 algoritmom je prikazano na slici 6.15.

```
F3_sakrij(slika, poruka, ključ){
    bit_info = bitovi(RC4(poruka, ključ));
    vel_por = bitovi(duljina(bit_info)); //veličina je spremljena u 16 bita
    stego = Kvantizator(DCT2D(slika));
    Za(i = 0; i < 16;){
        DCT_koef = DCT_koeficijent(stego, 'Sljedeći', 'AC', '!= 0');
        Ako(LSB(DCT_koef) != vel_por[i]) DCT_koef -= sign(DCT_koef);
        Ako(DCT_koef != 0) i++;
    }
    Za(i = 0; i < duljina(bit_info);){
        DCT_koef = DCT_koeficijent(stego, 'Sljedeći', 'AC', '!= 0');
        Ako(LSB(DCT_koef) != bit_info[i]) DCT_koef -= sign(DCT_koef);
        Ako(DCT_koef != 0) i++;
    }
    stego= Huffmanov_koder(stego);
    Vrati stego;
}
```

Slika 6.15 Pseudokod skrivanja poruke F3 algoritmom

U algoritmu na slici 6.15 poruka se skriva slijedno, a steganografski ključ se koristi samo za RC4 algoritam. Izbacivanjem RC4 algoritma se dobiva algoritam čiste steganografije (*pure steganography*) [1]. Algoritmi čiste steganografije su karakteristični po tome što ne zahtijevaju tajnu informaciju za razmjenu poruka. U tu skupinu algoritama spada i JSTEG algoritam. Valja primijetiti da se u F3 algoritmu na slici 6.15 koriste samo AC koeficijenti. Općenito je dobro izbjegavati DC koeficijente jer oni nose zapis o srednjoj vrijednosti bloka veličine 8×8. Algoritmi F4 i F5 također koriste samo AC koeficijente. Algoritam dohvaćanja poruke skrivene F3 algoritmom je prikazan na slici 6.16.

```
F3_dohvati(stego, ključ){
    stego= Huffmanov_dekoder(stego);
    Za(i = 0; i < 16; i++){
        DCT_koef = DCT_koeficijent(stego, 'Sljedeći', 'AC', '!= 0');
        vel_por[i] = LSB(DCT_koef);
    }
    duljina_poruke = int(vel_por);
    Za(i = 0; i < duljina_poruke; i++){
        DCT_koef = DCT_koeficijent(stego, 'Sljedeći', 'AC', '!= 0');
        bit_info[i] = LSB(DCT_koef);
    }
    poruka = RC4(char(bit_info), ključ);
    Vrati poruka;
}
```

Slika 6.16 Pseudokod dohvaćanja skrivene poruke (F3)

Na slici 6.17 je prikazana originalna slika i stego slika dobivena F3 algoritmom. U sliku je skrivena poruka veličine 100% kapaciteta.

Slika 6.17 Originalna slika i stego slika (F3)

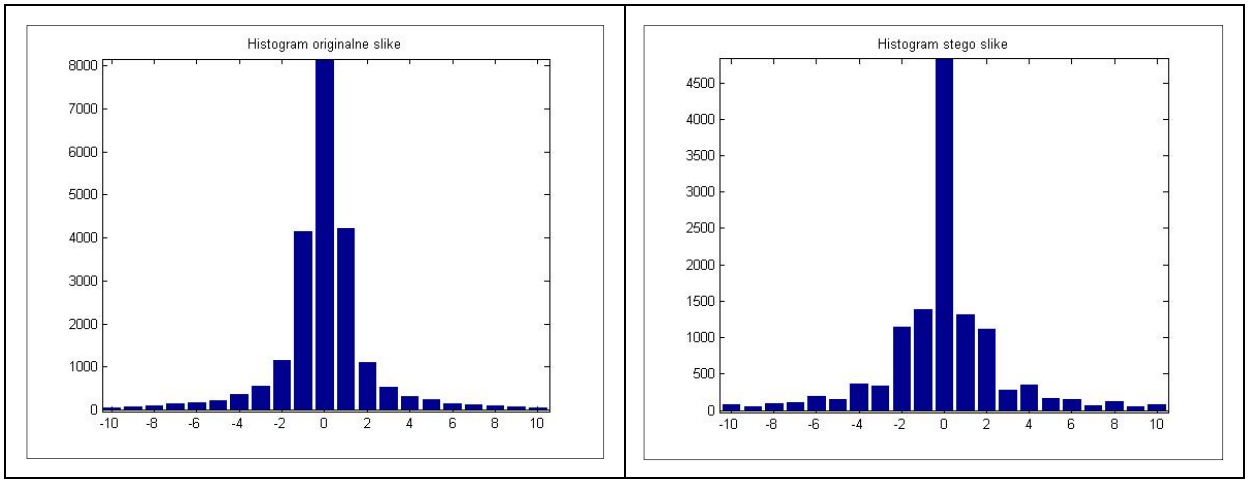
Pri računanju kapaciteta originalne slike za F3 algoritam potrebno je uzeti u obzir sažimanje. U radu se za određivanje kapaciteta koristi relacija:

$$C = \sum_{i \neq 0} h_i - 0.8 \cdot (h_{-1} + h_1) \tag{6.14}$$

U izrazu (6.14) oznake oblika h_i predstavljaju vrijednosti histograma AC koeficijenata. Umanjitelj $0.8 \cdot (h_{-1} + h_1)$ odražava utjecaj sažimanja na kapacitet predviđen za smještanje poruke. Faktor 0.8 je određen na temelju testiranja provedenog nad 5 slika: tražio se

minimalni faktor iz skupa {0.5, 0.55, 0.6,…,0.95, 1} za kojeg se može sakriti slučajna poruka veličine *C*. Kao i kod prethodnih algoritma, provedeno je testiranje količine podataka koju se može sakriti F3 algoritmom u zadani objekt nositelj. Rezultat testiranja pokazuje da je relativni kapacitet 9.63% uz standardnu devijaciju jednaku 3.98%.

Na slici 6.18 su prikazani histogrami DCT koeficijenata slika 6.17. Radi bolje preglednosti je uzet samo dio histograma i on je uvećan. Iz slike 6.18 se mogu uočiti neke karakteristike. Proces skrivanja kod F3 algoritma nema tendenciju izjednačavanja vrijednosti h_{2i} i h_{2i+1} u histogramu DCT koeficijenata. Stoga hi kvadrat test nije uspješan u analizi slike. Osim toga, algoritam F3 čuva simetričan oblik histograma.



Slika 6.18 Histogram DCT koeficijenata slike nositelja i stego slike

6.2.2 Napad na F3 algoritam

Za histogram DCT koeficijenata normalne slike uobičajeno vrijede relacije:

$$\begin{cases} h_i > h_{i+1}, i > 0 \\ h_i > h_{i-1}, i < 0 \end{cases} \tag{6.15}$$

Potvrdu toga se može vidjeti i u histogramu slike nositelja prikazanom na slici 6.18. Promotri li se detaljnije histogram stego slike prikazan na slici 6.18, može se uočiti da za velik broj vrijednosti *i* vrijedi:

$$\begin{cases} h_{2i} > h_{2i-1}, i > 0 \\ h_{2i} > h_{2i+1}, i < 0 \end{cases} \tag{6.16}$$

Kako relacije (6.16) nisu istinite u slučaju slike nositelja, histogram stego slike se može iskoristiti u detekciji skrivene poruke.

Ovdje je predložen algoritam steganalize opisan sljedećim koracima:

1. AC koeficijenti se uzimaju slijedno: prvo njih 1%, zatim 2%, i tako dalje. Uvijek se uzimaju od iste početne pozicije, tj. prvog od AC koeficijenta.
2. Na svakoj grupi AC koeficijenata se izračuna histogram DCT koeficijenata.
3. Za svaku grupu AC koeficijenata se na temelju histograma odredi relativna mjera nepravilnosti. Relativna mjera nepravilnosti se određuje na sljedeći način:
   3.1 Za svaki $i > 0$ se izračuna mjera δ_i formulom (6.17).

$$\delta_i = h_{2i} - \ h_{2i-1} \tag{6.17}$$

3.2 Za svaki $i < 0$ se izračuna mjera δ_i formulom (6.18).

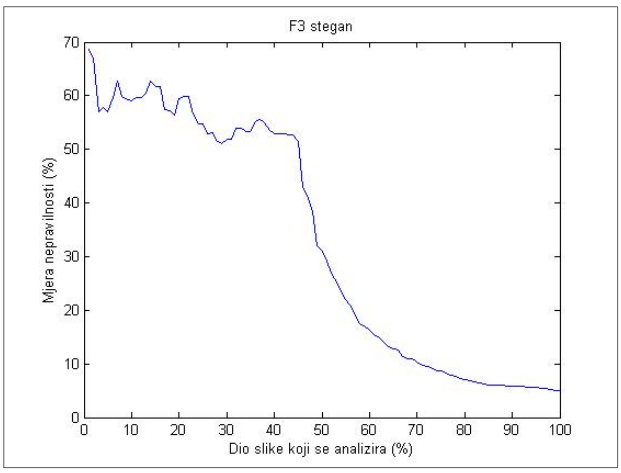
$$\delta_i = h_{2i} - \ h_{2i+1} \tag{6.18}$$

3.3 Relativna mjera nepravilnosti Δ se izračuna formulom (6.19).

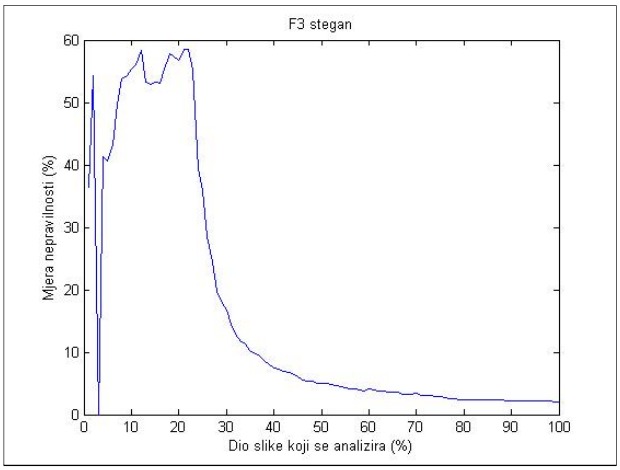
$$\Delta = \frac{\sum_{i \neq 0, \delta_i > 0} \delta_i}{\sum_{i \neq 0} \delta_i} \tag{6.19}$$

4. Redom se prolazi kroz grupe dok se ne nađe grupa čija je relativna mjera nepravilnosti manja od zadanog praga. Relativna veličina poruke je određena udjelom koji zadnja grupa veća od praga zauzima u slici.

Kako bi se odredila vrijednost praga, provedeno je testiranje na 5 slika. Pokazalo se da prag vrijednosti 0.3 daje najbolje rezultate. Opisani algoritam će se u radu referencirati kao *F3_stegan*.



Slika 6.19 Ovisnost mjere nepravilnosti o udjelu slike (1)



Slika 6.20 Ovisnost mjere nepravilnosti o udjelu slike (2)

Na slici 6.19 je prikazana ovisnost relativne mjere nepravilnosti o veličini dijela slike koji se analizira. U sliku je skrivena poruka veličine 50% kapaciteta, a procjena algoritma je bila ista. Na slici 6.20 je u istu sliku spremljena poruka veličine 25% kapaciteta, a procjena algoritma je također bila ista. Valja uočiti skokovitu promjenu na početku grafa. Taj singularitet bi po četvrtom koraku trebao prouzročiti krivu procjenu. Ipak, u algoritam je ugrađena provjera singulariteta, pa je procjena ipak ispala točna.

Kako bi se relativna veličina poruke preračunala u apsolutnu veličinu poruke potrebno je poznavati kapacitet slike nositelja. U prije opisanim algoritmima steganalize, stego slika ima isti kapacitet kao i slika nositelj pa nije bilo problema u određivanju apsolutne veličine poruke. Formula (6.14) za sliku nositelja i za stego sliku dobivenu F3 algoritmom ne daje iste rezultate. Razlog tome je sažimanje. No kapacitet originalne slike se može procijeniti ako je točna procjena relativne veličine poruke. Ukoliko je relativna veličina poruke jednaka p, broj DCT koeficijenata koji prijeđu u nulu u prosjeku je jednak $0.8 \cdot p \cdot (h_{-1} + h_1)$ gdje su h_i vrijednosti histograma AC koeficijenata za originalnu sliku. Dakle, broj nula u stego slici je jednak:

$$h_0^* = h_0 + 0.8 \cdot p \cdot (h_{-1} + h_1) \tag{6.20}$$

Ovdje se s h_i^* označavaju vrijednosti histograma AC koeficijenata za stego sliku. Broj dvojki koje prijeđu u jedinicu je jednak $0.5 \cdot p \cdot h_2$, a broj minus dvojki koje prijeđu u minus jedinicu je jednak $0.5 \cdot p \cdot h_{-2}$. Iz toga slijede jednakosti:

$$\begin{aligned} h_{\pm 1}^* &= (1 - 0.8 \cdot p) \cdot h_{\pm 1} + 0.5 \cdot p \cdot h_{\pm 2} \\ h_{-1}^* + h_{+1}^* &= (1 - 0.8 \cdot p) \cdot (h_{-1} + h_1) + 0.5 \cdot p \cdot (h_{-2} + h_2) \end{aligned} \tag{6.21}$$

Relacija (6.21) se može napisati i kao:

$$(h_{-1} + h_1) = \frac{h_{-1}^* + h_{+1}^* - 0.5 \cdot p \cdot (h_{-2} + h_2)}{1 - 0.8 \cdot p} \tag{6.22}$$

Na sličan način se dobiva da za $i > 1$ vrijedi:

$$\begin{aligned} h_{\pm i}^* &= (1 - 0.5 \cdot p) \cdot h_{\pm i} + 0.5 \cdot p \cdot h_{\pm (i+1)} \\ h_{-i}^* + h_i^* &= (1 - 0.5 \cdot p) \cdot (h_{-i} + h_i) + 0.5 \cdot p \cdot (h_{-(i+1)} + h_{i+1}) \\ (h_{-i} + h_i) &= \frac{h_{-i}^* + h_i^* - 0.5 \cdot p \cdot (h_{-(i+1)} + h_{i+1})}{1 - 0.5 \cdot p} \end{aligned} \tag{6.23}$$

Izraz (6.14) se može napisati u obliku:

$$C = \sum_{\substack{i \neq 0 \\ i \neq -1 \\ i \neq 1}} (h_i - 0.5ph_i + 0.5ph_i) + \sum_{i=\{-1,1\}} (h_i - 0.8ph_i + 0.8ph_i) - 0.8 \cdot (h_{-1} + h_1) \tag{6.24}$$

Bez velike pogreške se prva suma u izrazu (6.24) može promatrati kao beskonačna. U tom slučaju se izraz (6.24) može napisati kao:

$$C \approx \sum_{\substack{i\neq 0\\ i\neq -1\\ i\neq 1}} \left((1-0.5p)h_i + 0.5ph_{i+sgn(i)}\right) + \sum_{i=\{-1,1\}} \left((1-0.8p)h_i + 0.5ph_{2\cdot sgn(i)}\right) - 0.8(1-p)\cdot(h_{-1}+h_1) \tag{6.25}$$

Kombinirajući jednadžbe (6.21)-(6.25) dobiva se relacija:

$$C \approx \sum_{i\neq 0} h_i^* - \frac{0.8\cdot(1-p)}{1-0.8\cdot p}\cdot \sum_{i=1}^{\max(abs(AC))} (-1)^{i-1} q^{i-1}\Lambda_i$$
$$q = \frac{0.5\cdot p}{1-0.5\cdot p} \tag{6.26}$$
$$\Lambda_i = h_{-i}^* + h_i^*$$

U izrazu (6.26) su uvedene pokrate q i Λ_i. Izraz (6.26) govori kako procijeniti kapacitet originalne slike iz histograma stego slike. Druga suma u izrazu (6.26) se može aproksimirati s prvih nekoliko članova. U radu su uzeta prva tri člana pa formula po kojoj se procjenjuje kapacitet ima oblik:

$$C \approx \sum_{i\neq 0} h_i^* - \frac{0.8\cdot(1-p)}{1-0.8\cdot p}\cdot(\Lambda_1 - \mathrm{q}\Lambda_2 + q^2\Lambda_3) \tag{6.27}$$

Nakon što je kapacitet procijenjen, apsolutna duljina poruke se računa množenjem relativne duljine poruke s kapacitetom.

Mjerenje uspješnosti detekcije F3_stegan algoritma je provedeno kao s prethodnim algoritmima steganalize opisanim u ovom poglavlju uz razliku što je u ovom slučaju za skrivanje poruke korišten F3 algoritam. Točna procjena kapaciteta poruke, provjera singulariteta kod promatranja relativne mjere nepravilnosti te dobar odabir praga uzrok su relativno dobrim rezultatima kako u procjeni veličine, tako i u detekciji skrivene poruke.

Tablica 6.7 Uspješnost procjene duljine poruke (F3_stegan)

Veličina skrivene poruke	25%	50%	75%	**Ukupni prosjek**
Apsolutna razlika (%)	2.12% (1.67%)	3.8% (2.51%)	3.4% (3.24%)	3.11% (2.6%)
Relativna razlika (%)	8.52% (6.68%)	7.6% (5.0%)	4.56% (4.33%)	6.9% (5.57%)

Tablica 6.8 Uspješnost detekcije skrivene poruke (F3_stegan)

Točnost	**Preciznost**	**Odziv**	**Specifičnost**	**F**
0.9	0.83	1.0	0.8	0.91

6.2.3 F4 algoritam

F4 algoritam je u mnogome sličan F3 algoritmu, a od njega se razlikuje po načinu umetanja bita poruke u odgovarajući AC koeficijent. Kod F3 algoritma se koeficijentu smanji apsolutna vrijednost ukoliko njegov najmanje značajan bit ne odgovara njemu namijenjenom bitu poruke. Ako se promatrani AC koeficijent označi s AC_{koef}, a bit poruke s x, onda se umetanje F3 algoritmom može opisati izrazom:

$$AC_{koef} = AC_{koef} - sgn(AC_{koef}) \cdot [LSB(AC_{koef}) \oplus bit] \tag{6.28}$$

F4 algoritam pozitivne koeficijente tretira na isti način kao i F3 algoritam. Drugim riječima, bit poruke smanjuje vrijednost koeficijenta samo ako najmanje značajan bit koeficijenta ne odgovara bitu poruke. U slučaju negativnih koeficijenata vrijedi obrnuto pravilo: bit poruke povećava vrijednost koeficijenta samo ako najmanje značajan bit koeficijenta odgovara bitu poruke. Spomenuto se može opisati izrazom:

$$AC_{koef} = AC_{koef} - sgn(AC_{koef}) \cdot [LSB(AC_{koef}) \oplus bit \oplus (AC_{koef} < 0)] \tag{6.29}$$

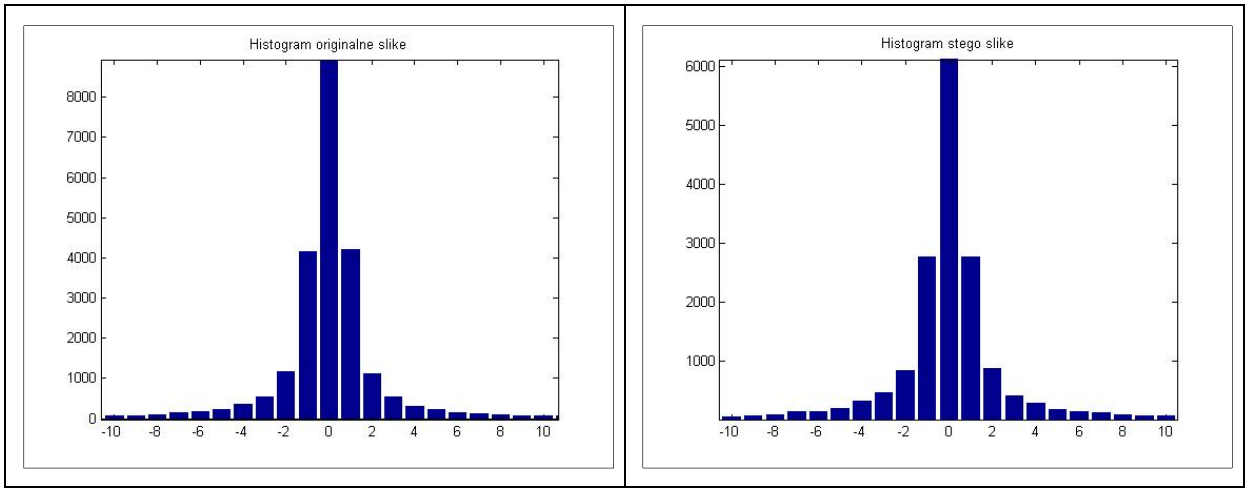
Dakle, kako bi se dobio F4 algoritam iz pseudokoda na slici 6.15, sve što treba napraviti je zamijeniti linije `Ako(LSB(DCT_koef) != vel_por[i]) DCT_koef -= sign(DCT_koef);` i `Ako(LSB(DCT_koef) != bit_info[i]) DCT_koef -= sign(DCT_koef);` s izrazom (6.29). Kod faze dohvaćanja treba pripaziti na svojstvo da je u slučaju negativnog koeficijenta bit poruke jednak komplementu najmanje značajnog bita koeficijenta.

Slika 6.21 Originalna slika i stego slika (F4)

Na slici 6.21 je prikazana originalna slika i stego slika dobivena F4 algoritmom. U sliku je skrivena poruka veličine 100% kapaciteta. Kapacitet se mjeri na jednak način kao i kod F3 algoritma uz razliku što je testom dobivena vrijednost faktora 0.55, a ne 0.8. U principu je dovoljno uzeti vrijednost faktora nešto veću od 0.5. Rezultat mjerenja relativnog kapaciteta provedenog na 5 slika iznosi 11.86% uz standardnu devijaciju jednaku 4.75%.

Na slici 6.22 su prikazani histogram originalne slike i histogram stego slike. Kao i kod F3 algoritma, proces skrivanja nema tendenciju izjednačavanja vrijednosti h_{2i} i h_{2i+1} u histogramu DCT koeficijenata pa hi kvadrat test nije uspješan u analizi slike dobivene F4 algoritmom. Također, i F4 algoritam čuva simetričan oblik histograma. Bitno je naglasiti da F4 algoritam

čuva i monotonost histograma pa F3_stegan nije uspješan u analizi stego slika dobivenih F4 algoritmom. To se može i matematički pokazati.



Slika 6.22 Histogram DCT koeficijenata slike nositelja i stego slike

Histogram DCT koeficijenata se može promatrati kao slučajna varijabla. Neka je histogram DCT koeficijenata originalne slike opisan slučajnom varijablom X, a histogram DCT koeficijenata stego slike opisan slučajnom varijablom Y. Vjerojatnost da X poprimi vrijednost x je označeno s $P(X = x)$, a vjerojatnost da Y poprimi vrijednost y je označeno s $P(Y = y)$. Kako bi postupak bio valjan i za F5 algoritam, ovdje će se pretpostaviti da se skriva poruka relativne veličine p i to raštrkano po slici (iako F4 algoritam skriva slijedno). Dakle, p koeficijenata je korišteno pri skrivanju poruke. Promotrimo što se događa kada se u sliku skriva najveća moguća poruka. Zbog monotonosti histograma originalne slike (relacija (6.15)) vrijedi:

$$\begin{aligned} P(X=0) > P(X=1) > P(X=2) > \ldots > P(X=n) \\ P(X=0) > P(X=-1) > P(X=-2) > \cdots > P(X=-n) \end{aligned} \tag{6.30}$$

Osim toga, oblik histograma originalne slike upućuje na relacije [18]:

$$\begin{aligned} P(X=0) - P(X=1) > \cdots > P(X=n) - P(X=n+1) > \cdots > 0 \\ P(X=0) - P(X=-1) > \cdots > P(X=-n) - P(X=-n-1) > \cdots > 0 \end{aligned} \tag{6.31}$$

Ovdje se ne zahtjeva da relacije (6.30) i (6.31) uistinu vrijede, već se nastoji pokazati da ih F4 algoritam neće narušiti ukoliko su zadovoljene. Uz pretpostavku da se bitovi poruke ravnaju prema uniformnoj distribuciji, vrijednosti $P(Y = y)$ su jednake:

$$\begin{aligned} P(Y=n) &= \left(1-\frac{p}{2}\right)P(X=n) + \frac{p}{2}P(X=n+1) \\ P(Y=n+1) &= \left(1-\frac{p}{2}\right)P(X=n+1) + \frac{p}{2}P(X=n+2) \\ P(Y=n+2) &= \left(1-\frac{p}{2}\right)P(X=n+2) + \frac{p}{2}P(X=n+3) \end{aligned} \tag{6.32}$$

Oduzimanjem prve i druge jednadžbe te druge i treće jednadžbe dobivaju se izrazi:

$$P(Y=n)-P(Y=n+1) = \left(1-\frac{p}{2}\right)\left(P(X=n)-P(X=n+1)\right)+\frac{p}{2}\left(P(X=n+1)-P(X=n+2)\right)$$
$$P(Y=n+1)-P(Y=n+2) = \left(1-\frac{p}{2}\right)\left(P(X=n+1)-P(X=n+2)\right)+\frac{p}{2}\left(P(X=n+2)-P(X=n+3)\right) \quad (6.33)$$

Uvažavajući izraze (6.30) dolazi se do relacija:

$$P(Y=n)-P(Y=n+1)>0 \Rightarrow P(Y=n)>P(Y=n+1)$$
$$P(Y=n+1)-P(Y=n+2)>0 \Rightarrow P(Y=n+1)>P(Y=n+2) \quad (6.34)$$

Dakle, vrijedi $P(Y = n) > P(Y = n+1) > P(Y = n+2)$. Na isti način se dokazuju nejednakosti $P(Y = -n) > P(Y = -n-1) > P(Y = -n-2)$. Zbog relacije (6.31) vrijedi:

$$\left(1-\frac{p}{2}\right)\left(P(X=n)-P(X=n+1)\right)>\left(1-\frac{p}{2}\right)\left(P(X=n+1)-P(X=n+2)\right)$$
$$\frac{p}{2}\left(P(X=n+1)-P(X=n+2)\right)>\frac{p}{2}\left(P(X=n+2)-P(X=n+3)\right) \quad (6.35)$$

Prema tome, desna strana prve jednadžbe u (6.33) je veća od desne strane druge jednadžbe u (6.33) pa se i lijeve strane tih jednadžbi tako odnose. Dakle, vrijedi:

$$P(Y=n)-P(Y=n+1)>P(Y=n+1)-P(Y=n+2) \quad (6.36)$$

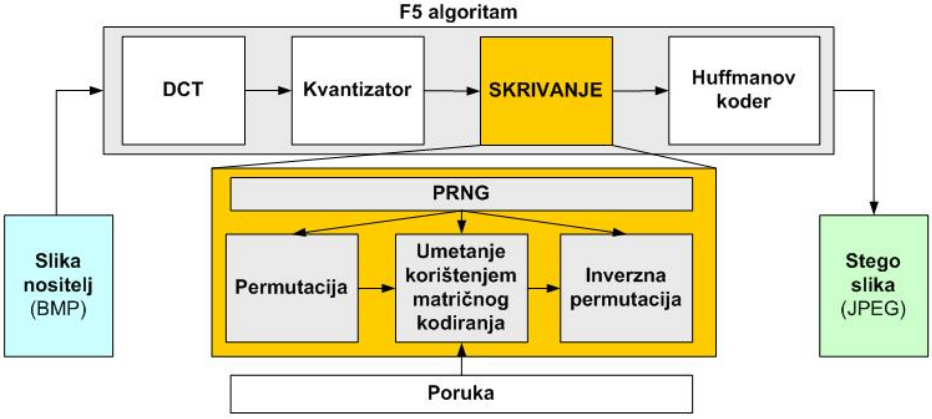
Na isti način se dobije odnos:

$$P(Y=-n)-P(Y=-n-1)>P(Y=-n-1)-P(Y=-n-2) \quad (6.37)$$

Time je pokazano da F4 čuva monotonost histograma.

6.2.4 F5 algoritam

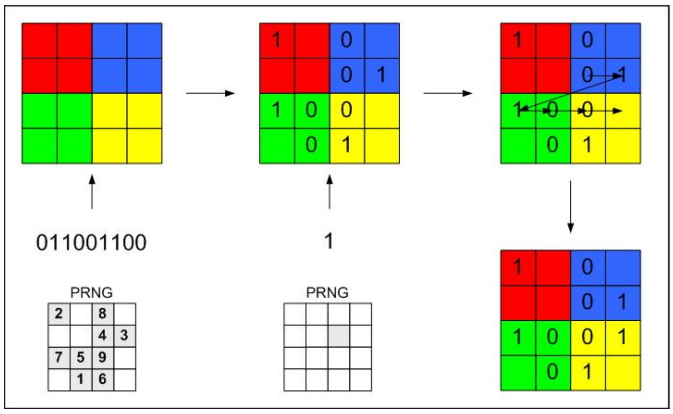
U osnovi *F5* algoritma se nalazi F4 algoritam. Unaprjeđenje se ogleda u korištenju dviju tehnika: permutacijsko raspršenje (*permutative stradling*) i matrično kodiranje (*matrix encoding*). Proces skrivanja je prikazan dijagramom na slici 6.23.



Slika 6.23 Proces skrivanja kod F5 algoritma

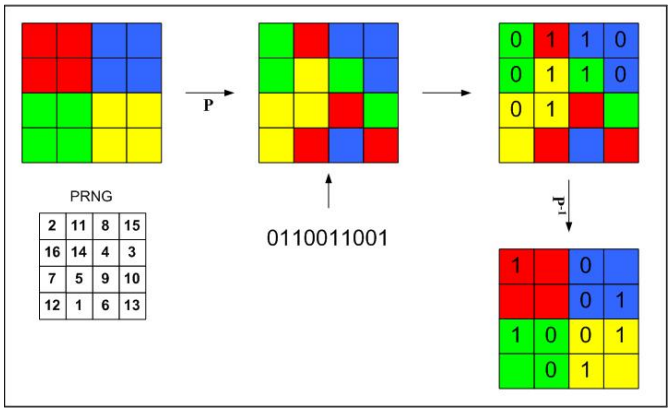
6.2.4.1 *Permutacijsko raspršenje*

Raspršeno skrivanje je otpornije na napade tehnika steganalize. Brzina skrivanja uvelike ovisi o načinu postizanja raspršenog skrivanja posebice kada se skriva poruka veličine kapaciteta slike nositelja. Ukoliko se relativno velika poruka skriva tako da se rade slučajne šetnje (*random walks*) s traženjem slobodnih koeficijenata, proces skrivanja može značajno usporiti pri samom kraju jer se pretražuje veći dio slike. Stoga se u F5 algoritmu koristi tehnika permutacijskog raspršenja. Na slici 6.24 je ilustrirano skrivanje slučajnom šetnjom, a na slici 6.25 je ilustrirana tehnika permutacijskog raspršenja



Slika 6.24 Skrivanje slučajnom šetnjom

U F5 algoritmu se poruka zapravo skriva slijedno, no prije samog umetanja se slika permutira permutacijom koja ovisi o steganografskom ključu. Nakon umetanja se obavlja inverzna permutacija. Time se postiže da su bitovi poruke, iako spremljeni slijedno, raspršeni po cijeloj slici. Budući da se pri umetanju ne rade slučajne šetnje s traženjem slobodnih koeficijenata, već se poruka skriva slijedno, ovakav način skrivanja neće s porastom veličine poruke uzrokovati značajno usporavanje procesa skrivanja. U slučaju manjih poruka je ovakav način skrivanja nešto sporiji.



Slika 6.25 Skrivanje permutacijskim raspršenjem

6.2.4.2 *Matrično kodiranje*

Kako bi se povećala učinkovitost umetanja, F5 algoritam koristi matrično kodiranje. Bez matričnog kodiranja bi, uz zanemarivanje sažimanja i pretpostavku da ulazna poruka prati uniformnu razdiobu bitova, učinkovitost umetanja bila jednaka 2 bita po promjeni. Sažimanje ima učinak smanjivanja učinkovitosti umetanja. Matrično kodiranje je najjednostavnije

shvatiti na primjeru koji je preuzet iz [18]. Neka se dva bita x_1 i x_2 žele spremiti koristeći tri promjenjiva bita a_1, a_2 i a_3. Promjene se odvijaju prema izrazima:

$$\begin{aligned} x_1 = a_1 \oplus a_3 \ \wedge \ x_2 = a_2 \oplus a_3 \ &\Rightarrow \ nema\ promjene \\ x_1 \neq a_1 \oplus a_3 \ \wedge \ x_2 = a_2 \oplus a_3 \ &\Rightarrow \ a_1 = a_1 \oplus 1 \\ x_1 = a_1 \oplus a_3 \ \wedge \ x_2 \neq a_2 \oplus a_3 \ &\Rightarrow \ a_2 = a_2 \oplus 1 \\ x_1 \neq a_1 \oplus a_3 \ \wedge \ x_2 \neq a_2 \oplus a_3 \ &\Rightarrow \ a_3 = a_3 \oplus 1 \end{aligned} \tag{6.38}$$

Dakle, ulazni bitovi se kodiraju uz pomoć postojećih bitova pri čemu se vrši najviše jedna promjena. Korisno je uvesti funkciju sažimanja:

$$f(a_1, a_2, a_3) = (a_1 \oplus a_3, a_2 \oplus a_3) \tag{6.39}$$

Uz pomoć nje se može opisati operacija vađenja poruke izrazom:

$$(x_1, x_2) = f(a_1, a_2, a_3) \tag{6.40}$$

Osim toga, prilikom skrivanja poruke funkcija sažimanja određuje je li potrebno vršiti ikakvu promjenu i, ako je potrebno, koji bit se mijenja. Za to se koristi izraz (6.41). Veličina *s* se zove *sindrom* i dana je u bitovima. Ukoliko je njena dekadska vrijednost 0, tada nema promjene. Inače njena dekadska vrijednost određuje indeks bita koji je potrebno mijenjati.

$$s = f(a_1, a_2, a_3) \oplus (x_1, x_2) \tag{6.41}$$

Za gornji primjer se može kazati da je matrični kod (1, 2, 2^2-1), tj. kod s Hammingovom udaljenosti 1, veličinom ulaznih riječi 2 i veličinom kodnih riječi 3. Općenito, F5 algoritam koristi matrični kod (1, k, 2^k-1). Uz korištenje vektorskog zapisa, funkcija sažimanja se može definirati izrazom (6.42) gdje $\mathbf{y}_i$ označava k dimenzionalni vektor čije su vrijednosti bitovi binarnog zapisa brojača i (primjerice, za i = 3, $\mathbf{y}_i$ = (1, 1, 0)).

$$f(\mathbf{a}) = \oplus_{i=1}^{2^k-1} a_i \cdot \mathbf{y}_i \tag{6.42}$$

Na isti način se može modificirati izraz za sindrom:

$$\mathbf{s} = f(\mathbf{a}) \oplus \mathbf{x} \tag{6.43}$$

Neka se pri skrivanju poruke koristi matrični kod (1, k, 2^k-1). Također, neka je zanemareno sažimanje. Ako je dan vektor bitova **x**, vjerojatnost da vektor bitova **a** neće biti promijenjen, tj. da **a** kodira upravo **x**, iznosi $1/2^k$. *Gustoća promjene* (*change density*), koja mjeri relativni broj promijenjenih bitova, je dana izrazom:

$$D(k) = \left(1 - \frac{1}{2^k}\right) \cdot \frac{1}{2^k - 1} = \frac{1}{2^k} \tag{6.44}$$

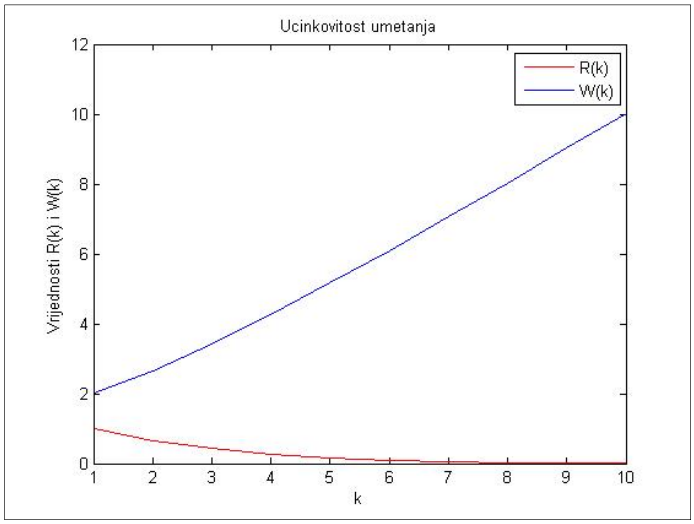
Frekvencija umetanja (*embedding rate*), tj. omjer broja bitova poruke i broja DCT koeficijenata namijenjenih skrivanju poruke, iznosi:

$$R(k) = \frac{k}{2^k - 1} \tag{6.45}$$

Učinkovitost umetanja se definira kao omjer frekvencije umetanja i gustoće umetanja, a predstavlja prosječan broj bitova koji se skriva pri jednoj promjeni. Učinkovitost je dana izrazom (6.46). Iz izraza slijedi da je učinkovitost umetanja uvijek veća od *k*.

$$W(k) = \frac{R(k)}{D(k)} = \frac{2^k}{2^k - 1}k \tag{6.46}$$

Na slici 6.26 je prikazana ovisnost frekvencije umetanja i učinkovitosti umetanja o *k*. Iz slike se vidi da se visoka učinkovitost umetanja može postići samo za nisku frekvenciju umetanja, odnosno kada se skrivaju kratke poruke.



Slika 6.26 Ovisnost R(k) i W(k) o k

6.2.4.3 Koraci skrivanja

Koraci skrivanja su opisani pseudokodom na slici 6.27. Steganografski ključ se koristi kako bi se kriptirala poruka RC4 algoritmom, ali i za generiranje permutacije. Prije nego se krenu skrivati bitovi poruke potrebno je odrediti prikladan *k* kako bi učinkovitost umetanja bila najveća.

```
F5_sakrij(slika, poruka, ključ){
    stego = Permutacija(Kvantizator(DCT2D(slika)), ključ);

    k = Inicijalizacija(stego);
    n = 2^k – 1;

    bit_info = bitovi(RC4(poruka, ključ));
    vel_por = bitovi(duljina(bit_info)); //veličina je spremljena u 16 bita
    bit_k = bitovi(k); //spremljeno u 16 bita

    F4_skrivanje(&stego, vel_por);
    F4_skrivanje(&stego, bit_k);
    Matrično_kodiranje(&stego, bit_info, 'Slijedno', k, n);

    stego= Huffmanov_koder(Inverzna_permutacija(stego, ključ));
    Vrati stego;
}
```

Slika 6.27 Pseudokod skrivanja poruke F5 algoritmom

Kako bi se odredio *k*, potrebno je poznavati kapacitet slike nositelja. Kapacitet se prema [19] procjenjuje formulom:

$$C = \sum_{i \neq 0} h_i - 0.51 \cdot (h_{-1} + h_1) \tag{6.47}$$

U izrazu (6.47) oznake oblika h_i predstavljaju vrijednosti histograma AC koeficijenata. Vrijednost *k* se dobiva izjednačavanjem frekvencije umetanja s relativnom duljinom poruke (duljina poruke u odnosu na kapacitet), tj. iz jednakosti (6.48) pri čemu je *L* duljina poruke u bitovima.

$$\frac{k}{2^k - 1} = \frac{L}{C} \tag{6.48}$$

U procesu skrivanja se prvo sakriju duljina poruke i *k*. To se radi kao u slučaju F4 algoritma. Bitovi poruke se skrivaju korištenjem matričnog kodiranja. Skrivanje bitova poruke se može opisati sljedećim koracima [18], [19]:

1. Redom se čitaju blokovi od *k* bitova poruke. Svaki blok čini jedan binarni vektor. Ako duljina poruke u bitovima nije djeljiva s *k*, onda se zadnji blok nadopuni. Budući da se primatelju šalje duljina poruke i *k*, on će moći odbaciti višak bitova.
2. Nakon što je pročitan blok poruke, uzima se još neiskorištena grupa od *n* AC koeficijenata različitih od 0 i njihovi najmanje značajni bitovi se poredaju u vektor. U vektoru se prevrnu bitovi dobiveni od negativnih koeficijenata.
3. Prema (6.43) se određuje sindrom. Ukoliko je sindrom 0, ne mijenja se niti jedan koeficijent i nastavlja se sa sljedećim blokom poruke. U suprotnome sindrom pokazuje na DCT koeficijent čiju apsolutnu vrijednost treba smanjiti. Ako je promjenom došlo do sažimanja, ponavljaju se koraci 2 i 3 pri čemu se u koraku 2 uzima samo jedan novi AC koeficijent u zamjenu za onoga koji je poprimio vrijednost 0. Inače se nastavlja sa sljedećim blokom poruke.

Na slici 6.28 je prikazan pseudokod dohvaćanja skrivene pouke. Korak dohvaćanja bitova poruke koristi matrično dekodiranje. Redom se dohvaćaju grupe od *n* AC koeficijenata različitih od 0. Iz grupe se stvori binarni vektor tako da se izvuku najmanje značajni bitovi koeficijenata. U vektoru se prevrnu bitovi dobiveni od negativnih koeficijenata te se na temelju relacije (6.40) odredi blok od *k* bitova poruke. Spajajući blokove i odbacujući moguće viškove dobiva se kriptirana poruka. Primjenom RC4 algoritma se odredi izvorna poruka.

```
F5_dohvati(stego, ključ){
    stego= Permutacija(Huffmanov_dekoder(stego), ključ);

    duljina_poruke = int(F4_čitanje(16));
    k = int(F4_čitanje(16));
    n = 2^k – 1;

    bit_info = Mat_dekodiranje(stego, 'Slijedno', duljina_poruke, k, n);

    poruka = char(RC4(bit_info, ključ));
    Vrati poruka;
}
```

Slika 6.28 Pseudokod dohvaćanja skrivene poruke (F5)

Na slici 6.29 je prikazana originalna slika i stego slika dobivena F5 algoritmom. U sliku je skrivena poruka veličine 100% kapaciteta. F5 algoritam na jednak način modificira neprikladne AC koeficijente kao i F4 algoritam pa su i u ovom slučaju očuvana svojstva simetričnosti i monotonosti histograma. Stoga hi kvadrat test i F3_stegan algoritam nisu uspješni u analizi stego slika dobivenih F5 algoritmom. Za pokazati očuvanost monotonosti dovoljno je u dokazu napravljenom u odjeljku 6.2.3 veličinu *p/2* zamijeniti s gustoćom promijene *D*(*k*).

Slika 6.29 Originalna slika i stego slika (F5)

Rezultat mjerenja relativnog kapaciteta provedenog nad 5 slika iznosi 12.22% uz standardnu devijaciju jednaku 4.87%.

6.2.5 Napad na F5 algoritam

Iako se za steganalizu stego slika dobivenih F5 algoritmom može upotrijebiti modifikacija OG_stegan algoritma, u radu je implementiran napad opisan u [15] i [19].

Neka proces skrivanja mijenja *n* AC koeficijenata od njih *P* prikladnih za umetanja (različitih od 0). *Frekvencije promjene* (*change rate*) β je tada jednaka *n*/*P*. Ova veličina je ekvivalent veličini *D*(*k*) iz prethodnog odjeljka, no sada nije zanemareno sažimanje. Kao što je spomenuto u poglavlju o JPEG kompresiji, DCT transformacija slike se provodi nad blokovima veličine 8×8. Dakle, svaki DCT koeficijent ima svoju poziciju u odgovarajućem bloku. Neka je kod slike nositelja broj AC koeficijenata čija je apsolutna vrijednost *d* i koji se nalaze na poziciji (*u*, *v*) u blokovima slike označen s $h_{uv}(d)$. Ista veličina za stego sliku je ovdje označena s $H_{uv}(d)$. Očekivana vrijednost $H_{uv}(d)$ je jednaka:

$$\begin{cases} H_{uv}(0) = h_{uv}(0) + \beta \cdot h_{uv}(1), \\ H_{uv}(d) = (1-\beta) \cdot h_{uv}(d) + \beta \cdot h_{uv}(d+1), d > 0 \end{cases} \tag{6.49}$$

Ako je dana procjena $\hat{h}_{uv}(d)$ za veličinu $h_{uv}(d)$, tada se korištenjem izraza (6.49) može procijeniti β. Pri tome se veličine $H_{uv}(d)$ izračunaju temeljem stego slike. Procjena se obavlja minimizacijom kvadratne greške pri čemu je za *d* dovoljno uzeti 0 i 1 jer za te vrijednosti veličina $h_{uv}(d)$ doživljava najveće promijene. Matematički se procjena izražava kao:

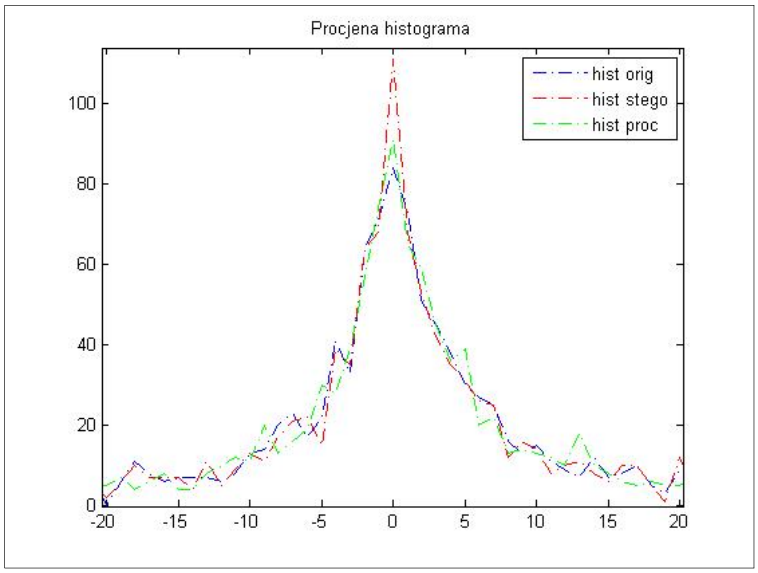
$$\beta_{uv} \approx arg\min_{\beta}\left[\left(H_{uv}(0) - \hat{h}_{uv}(0) - \beta \cdot \hat{h}_{uv}(1)\right)^2 + \left(H_{uv}(1) - (1-\beta) \cdot \hat{h}_{uv}(1) - \beta \cdot \hat{h}_{uv}(2)\right)^2\right] \tag{6.50}$$

Deriviranjem izraza u uglatim zagradama i izjednačavanjem s nulom (nužni uvjet ekstrema) dobiva se formula kojom se određuje β_{uv}:

$$\beta_{uv} \approx \frac{\left(H_{uv}(0) - \hat{h}_{uv}(0)\right) \cdot \hat{h}_{uv}(1) + \left(H_{uv}(1) - \hat{h}_{uv}(1)\right) \cdot \left(\hat{h}_{uv}(2) - \hat{h}_{uv}(1)\right)}{\hat{h}_{uv}^2(1) + \left(\hat{h}_{uv}(2) - \hat{h}_{uv}(1)\right)^2} \tag{6.51}$$

Veličina β se računa kao prosjek od β_{uv} pri čemu se razmatraju samo niže frekvencije, tj. za (u,v) se uzimaju vrijednosti iz skupa {(1,2),(2,1),(2,2)}. Naime, korištenjem viših frekvencija se može značajno narušiti točnost aproksimacije [19].

Točnost aproksimacije uvelike ovisi o procijenjenim vrijednostima $\hat{h}_{uv}(d)$. Za originalnu sliku h_{uv} predstavlja histogram apsolutnih vrijednosti AC koeficijenata na poziciji (u, v). Procjena tog histogama, tj. $\hat{h}_{uv}$, se dobiva na isti način kao i procjena histograma DCT koeficijenata kod OG_stegan algoritma. Dakle, stego sliku se prebaci u prostornu domenu te joj se odstrane prva četiri stupca. Tako dobivenu sliku se prebaci u kvantiziranu DCT domenu te se nad njom izračuna $\hat{h}_{uv}(d)$. Osim što je potrebno izračunati $\hat{h}_{uv}(d)$, za određivanje stvarne veličine poruke (veličine poruke u bitovima), potrebno je procijeniti i histogram slike $\hat{h}$. Na slici 6.30 je prikazan histogram originalne slike, histogram stego slike i procijenjeni histogram originalne slike za poziciju (1, 2).



Slika 6.30 Procjena histograma za poziciju (1,2)

Nakon što je izračunat β određuje se stvarna veličina poruke. Za to je potrebno procijeniti, prema gore opisanom postupku, histogram AC koeficijenata $\hat{h}$. Iz histograma se dobiva $\hat{h}(1)$ koji označava broj AC koeficijenata čija je apsolutna vrijednost 1. Koristeći isti histogram odredi se i P, tj. njegova približna vrijednost. Broj promjena n se može izraziti relacijom (6.52) gdje je m broj promjena uzrokovan skrivanjem bitova poruke, a s broj promjena uzrokovan sažimanjem.

$$n = m + s \tag{6.52}$$

Sažimanje se može dogoditi samo ako je apsolutna vrijednost koeficijenta jednaka 1. Stoga je vjerojatnost izbora koeficijenta koji može uzrokovati sažimanje jednaka $\hat{h}(1)/P$ pa je

očekivana vrijednost broja promjena uzrokovanih sažimanjem jednaka $n\cdot\hat{h}(1)/P$. Iz svega navedenog slijedi da je broj promjena uzrokovan skrivanjem bitova poruke jednak:

$$m = n\left(1 - \frac{\hat{h}(1)}{P}\right) \tag{6.53}$$

Kako učinkovitost umetanja predstavlja prosječan broj bitova koji se skriva pri jednoj promjeni, veličina skrivene poruke je jednaka:

$$\begin{aligned} M = W(k)\cdot m &= \frac{2^k}{2^k-1}\cdot k\cdot n\cdot\left(1-\frac{\hat{h}(1)}{P}\right) = \frac{2^k}{2^k-1}\cdot k\cdot\beta\cdot P\cdot\left(1-\frac{\hat{h}(1)}{P}\right) \\ &= \frac{2^k}{2^k-1}\cdot k\cdot\beta\cdot\left(P-\hat{h}(1)\right) \end{aligned} \tag{6.54}$$

Vrijednost za *k* se određuje na sličan način kako je to učinjeno prilikom skrivanja poruke. Kako bi se dobila vrijednost od *k*, potrebno je procijeniti kapacitet originalne slike. Iz izraza (6.47) slijedi da je kapacitet jednak:

$$C = P - 0.51\cdot\hat{h}(1) \tag{6.55}$$

Vjerojatnost da će se pri smještanju jednog bloka od *k* bitova poruke dogoditi promjena je jednaka $1\text{-}1/2^k$. Broj grupa u koji se skrivaju blokovi poruke je jednak $C/(2^k\text{-}1)$ pa je ukupni broj promjena nastalih umetanjem poruke jednak:

$$m = \frac{2^k-1}{2^k}\cdot\frac{C}{2^k-1} = \frac{C}{2^k} \tag{6.56}$$

Iz izraza (6.56) proizlazi jednakost po kojoj se iz *C* i *m* računa *k*:

$$k = log_2\left(\frac{C}{m}\right) \tag{6.57}$$

Iz svega navedenog slijedi da su koraci detekcije i procjene duljine skrivene poruke sljedeći:

1. Nad stego slikom se izračunaju vrijednosti $H_{uv}(d)$.
2. Stego sliku se prebaci u prostornu domenu te joj se odstrane prva četiri stupca. Tako dobivenu sliku se prebaci u kvantiziranu DCT domenu te se nad njom izračuna $\hat{h}_{uv}(d)$. Na temelju histograma AC koeficijenata se određuju $\hat{h}(1)$ i *P*.
3. Koristeći formulu (6.51) se izračunaju vrijednosti β_{uv} pri čemu se (u,v) uzimaju iz skupa $\{(1,2),(2,1),(2,2)\}$. Veličina β se računa kao prosjek od β_{uv}.
4. Prvo se izrazom (6.55) odredi kapacitet *C*, a zatim se izrazom (6.57) odredi *k*.
5. Izračuna se duljina poruke korištenjem izraza (6.54).

Opisani algoritam će se u radu referencirati kao *F5_stegan*.

Mjerenje uspješnosti detekcije je provedeno kao s prethodnim algoritmima ovog poglavlja uz razliku što je u ovom slučaju za skrivanje korišten F5 algoritam. Rezultati su prikazani u tablicama 6.9 i 6.10. F5_stegan algoritam otprilike jednako dobro procjenjuje duljinu poruke kao i OG_stegan algoritam. S druge strane je detekcija skrivene poruke dosta lošija kod F5_stegan algoritma. Iz rezultata se vidi da F5_stegan ima velik broj FP detekcija.

Tablica 6.9 Uspješnost procjene duljine poruke (F5_stegan)

Veličina skrivene poruke	25%	50%	75%	**Ukupni prosjek**
Apsolutna razlika (%)	4.11% (3.57%)	5.68% (4.06%)	9.8% (8.5%)	6.53% (6.18%)
Relativna razlika (%)	16.47% (14.32%)	11.38% (8.12%)	13.08% (11.34%)	13.64% (11.47%)

Tablica 6.10 Uspješnost detekcije skrivene poruke (F5_stegan)

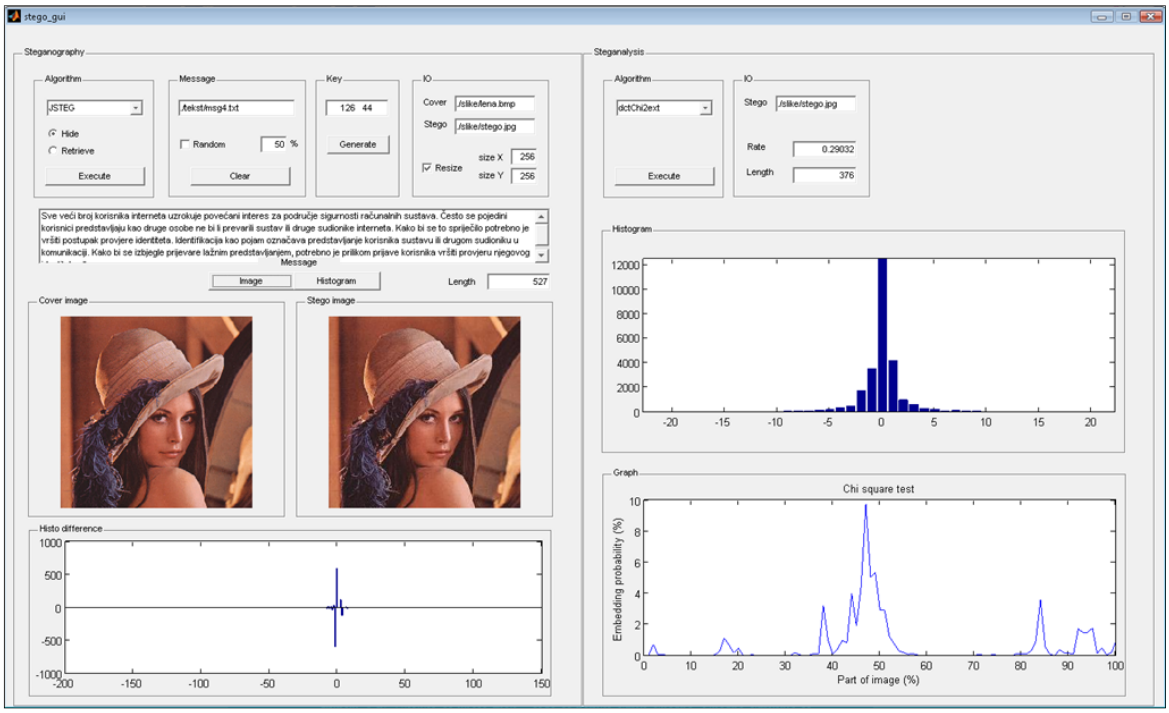
Točnost	**Preciznost**	**Odziv**	**Specifičnost**	**F**
0.65	0.64	0.7	0.6	0.67

Uspješnost procjene veličine skrivene poruke uvelike ovisi o procjeni histograma originalne slike. Velike pogreške u procjeni se mogu dogoditi u slučaju duple kompresije ili u slučaju kada u slici postoje pravilne strukture čija je karakteristična duljina usporediva s veličinom bloka. Načini kako u tim slučajevima smanjiti pogrešku su opisani u [19].

7. Praktični rad

7.1 Okruženje

Algoritmi opisani u prethodnim poglavljima su implementirani u razvojnom okruženju MATLAB. Za lakši rad s JPEG formatom slika se koristio *Matlab JPEG Toolbox* koji se može pronaći u [21]. U okviru rada je napravljeno grafičko sučelje prikazano na slici 7.1 i 7.2. Prozor grafičkog sučelja je podijeljen na dva dijela. Lijevi dio prozora je vezan za steganografske algoritme, a desni za metode steganalize.



Slika 7.1 Grafičko sučelje (1)

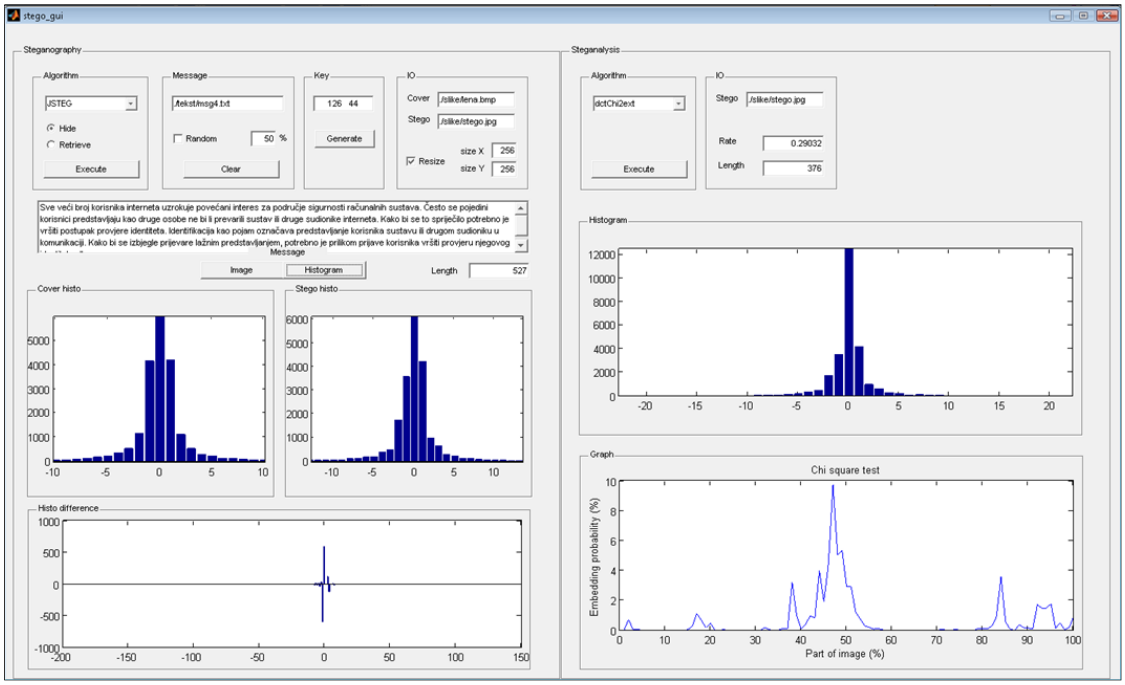
7.2 Opcije vezane uz steganografske algoritme

Steganografski algoritmi imaju dva načina korištenja: skrivanje poruke i dohvaćanje poruke. Način korištenja se određuje pritiskom na opciju *Hide* ili *Retrieve*. Pokretanje steganografskog algoritma se postiže pritiskom na *Execute*, a izvršava se onaj algoritam koji je izabran u padajućem izborniku. OutGuess, F3, F4 i F5 algoritmi se u padajućem izborniku referenciraju svojim nazivima. Naziv JSTEG zapravo spaja obični JSTEG i JSTEG_ext te se ovisno o zadanom ključu pokreće odgovarajući algoritam (ako ključ nije zadan pokreće se JSTEG, a inače se pokreće JSTEG_ext). sLSB je naziv za jednostavni LSB algoritam u prostornoj domeni, a ImgDowngrade je naziv za skrivanje slike u slici.

7.2.1 Skrivanje poruke

Tri parametra su bitna kod skrivanja: poruka, steganografski ključ i slika nositelj. Poruka se zadaje kao tekstualna datoteka: ime datoteke se unosi u kućicu namijenjenu unosu teksta pod grupom *Message*. Može se zadati i slučajna poruka relativne veličine. To se postiže pritiskom na opciju *Random* pod grupom *Message* te specificiranjem relativne veličine poruke u kućici

do. Poruka se ispisuje pritiskom na *Show*, a briše pritiskom na *Clear*. Svaki put kada se pokrene skrivanje poruke vrši se i njen ispis. Uz ispis poruke prikazuje se i njena duljina u oktetima. Steganografski ključ se zadaje u obliku dva broja. Ti brojevi se unose u kućicu pod grupom *Key*. Ukoliko se unese znak -, tada se poruka skriva slijedno. Slijedno skrivanje se podrazumijeva kod F3 i F4 algoritama, a opcionalno je za sLSB i JSTEG algoritme. Ključ se ne mora zadati jedino za sLSB i JSTEG. Generiranje slučajnog ključa se postiže pritiskom na *Generate*. Slika nositelj se zadaje u *IO* grupi: zadaje se naziv datoteke. Također se zadaje i naziv datoteke u koju će se spremiti stego slika. Veličina ulazne slike se može podesiti: potrebno je zadati veličinu u kućice *sX* i s*Y* te odabrati opciju *Resize*. Nakon što se poruka sakrije prikazuju se slika nositelj i stego slika u okvirima *Cover image*, odnosno *Stego image*. Ukoliko se klikne na *Histogram*, tada se okviri *Cover image* i *Stego image* zamijene okvirima *Cover histo* i *Stego histo* u kojima su prikazani histogrami slike nositelja i stego slike. Povratak na slike se postiže klikom na *Image*. U okviru *Histo difference* je prikazana razlika histograma slike nositelja i stego slike. Kod algoritama prilagođenih JPEG formatu se prikazuje histogram DCT koeficijenata.



Slika 7.2 Grafičko sučelje (2)

Za postupak skrivanja slike u slici opcije su nešto drugačije. Tajna slika se zadaje na mjestu gdje se uobičajeno zadaje datoteka poruke, veličina slike se uvijek podešava i ne koristi se nikakav ključ. Umjesto okvira *Cover image*, *Stego image*, *Cover histo* i *Stego histo*, koriste se okviri *Original* za sliku nositelja, *Secret (in)* za tajnu sliku koja se skriva, *Stego* za stego sliku i *Secret (out)* za tajnu sliku koja se dohvaća. Umjesto gumba *Histogram* i *Image*, koriste se gumbi *Image (in)* i *Image (out)*. *Image (in)* prikazuje okvire *Original* i *Secret (in)*, a *Image (out)* prikazuje okvire *Stego* i *Secret (out)*.

7.2.2 Dohvaćanje poruke

Za dohvatiti poruku potrebno je zadati dva parametra: stego sliku i steganografski ključ. Nakon pokretanja se ispisuje skrivena poruka i njena duljina. U slučaju skrivanja slike u slici se u okviru *Secret (out)* prikazuje tajna slika.

7.3 Opcije vezane uz algoritme steganalize

Pokretanje algoritma steganalize se postiže pritiskom na *Execute*, a izvršava se onaj algoritam koji je izabran u padajućem izborniku. Hi kvadrat test je referenciran s *dctChi2*, a prošireni hi kvadrat test je referenciran s *dctChi2ext*. Ostali algoritmi koriste nazive iz prethodnih poglavlja. Stego slika koja se analizira zadaje se imenom datoteke u *IO* grupi. Nakon što analiza završi ispisuje se procijenjena veličina skrivene poruke u kućici *Length*. Osim apsolutne veličine, ispisuje se i relativna veličina poruke u kućici *Rate*. Za F5_stegan algoritam se u kućici *Rate* ispisuje veličina β. U okviru *Histogram* se prikazuje histogram analizirane slike. Za algoritme koji operiraju nad JPEG slikama se u spomenutom okviru prikazuje histogram DCT koeficijenata. U tablici 7.1 je dan opis grafa koji se prikazuje u okviru *Graph* kod pojedinih algoritama steganalize.

Tablica 7.1 Grafovi koji se prikazuje u okviru Graph

Algoritam	***Graf***
PoV	*Vjerojatnost umetanja u ovisnosti o udjelu slike koji se analizira*
RS	*Vrijednosti R, S, R_{-1} i S_{-1}*
dctChi2	*Vjerojatnost umetanja u ovisnosti o udjelu slike koji se analizira*
dctChi2ext	*Vjerojatnost umetanja u ovisnosti o poziciji u slici*
OG_stegan	*Vrijednosti bločnosti: $B_s(0)$, $B_s(1)$, $B_0(0)$, $B_0(1)$, $B_1(1)$*
F3_stegan	*Mjera nepravilnosti u ovisnosti o udjelu slike koji se analizira*
F5_stegan	-

8. Zaključak

Steganografija predstavlja alternativu kriptografiji. Kako je uobičajeno veliki broj slika koje kolaju komunikacijskim kanalom, napadač teško može svaku sliku analizirati. U tom pogledu je steganografski sustav u prednosti pred kriptografskim sustavom jer napadač teško dolazi u doticaj s tajnom porukom. Posebice su zanimljivi steganografski sustavi koji koriste kriptirane poruke čime se dodatno poboljšava sigurnost sustava: čak i ako se izdvoji poruka ona je i dalje kriptirana i napadaču nerazumljiva. Negativna strana steganografije leži u činjenici što je za skrivanje poruke potrebno koristiti objekt nositelj koji uobičajeno nosi veću količinu podataka nego li sama poruka. Tehnike steganalize nastoje otkriti postoji li skrivena poruka unutar bezazlene informacije. Posebice su bitne tehnike steganalize koje kao podlogu koriste statističke testove. Takve tehnike, osim same detekcije skrivene poruke, često mogu odrediti i približnu veličinu poruke.

Testovi provedeni nad različitim algoritmima steganalize ukazuju na neka bitna svojstva. Prvo je činjenica da LSB supstitucijska tehnika ostavlja prepoznatljive statističke tragove. U prvom redu se to odnosi na izjednačavanje susjednih vrijednosti u histogramu slike. RS steganaliza pokazuje kako se LSB supstitucijom povećava zašumljenost slike. I u slučaju kada se ne koristi LSB supstitucija ili se njeni statistički tragovi prekrivaju, modifikacija slike može promijeniti njena statistička svojstva. Tako se kod JPEG slika mijenja mjera bločnosti. Ipak, skrivanjem relativno male količine podataka uspješnost detekcije opada. Posebice to vrijedi za algoritme steganalize koji su namijenjeni sofisticiranijim steganografskim metodama, poput OutGuess algoritma ili F5 algoritma.

Budući koraci uključuju razmatranje drugih oblika steganografije i steganalize, poput upotrebe tehnika raspršenog spektra u steganografiji ili tehnika nadziranog učenja u steganalizi.

Dodatak A: Mjere uspješnosti

U ovom dodatku će se ukratko definirati mjere uspješnosti korištene pri testiranju tehnika steganalize.

Neka su X, Y i Z varijable takve da vrijede odnosi: $0 \leq X \leq Z$, $0 \leq Y \leq Z$, $|X\text{-}Y| \leq X$, $|X\text{-}Y| \leq Y$. Apsolutnu razliku od X i Y u odnosu na referentnu varijablu Z ovdje definiramo, uz oznaku $A_Z(X, Y)$ ili kraće A, kao:

$$A = \frac{|X - Y|}{Z} \tag{A.1}$$

Relativnu razliku od X i Y u odnosu na varijablu Y ovdje definiramo, uz oznaku $R_Y(X, Y)$ ili kraće R, kao:

$$R = \frac{|X - Y|}{Y} \tag{A.2}$$

Gornje mjere uspješnosti se u radu koriste u analizi metoda steganalize pri procjenjivanju duljine skrivene poruke. Pri tome je varijabla X procijenjena duljina poruke, varijabla Y stvarna duljina poruke, a varijabla Z ukupni kapacitet namijenjen za skrivanje poruke.

Za analizu metoda steganalize u detekciji skrivene poruke koriste se mjere uspješnosti koje se mogu pronaći u [10]. Tablica zabune daje odnos između procijenjenog stanja i stvarnog stanja i ima oblik prikazan tablicom A.1. Mjera *TP* (*True Positive*) broji koliko se puta procijenjeno pozitivno i stvarno pozitivno poklopilo. Mjera *FP* (*False Positive*) broji koliko je puta procjena bila pozitivna, a trebala bi biti negativna. Mjera *FN* (*False Negative*) broji koliko je puta procjena bila negativna, a trebala je biti pozitivna. Mjera *TN* (*True Negative*) broji koliko se puta procijenjeno negativno i stvarno negativno poklopilo.

Tablica A.1 Tablica zabune

	Stvarno stanje		
Procjena		*Pozitivno*	*Negativno*
	Pozitivno	*TP*	*FP*
	Negativno	*FN*	*TN*

Pomoću mjera TP, FP, FN i TN se mogu definirati sljedeće mjere uspješnosti:

- Točnost (*Accuracy*) je udio točnih procjena:

$$Točnost = \frac{TP + TN}{TP + TN + FP + FN} \tag{A.3}$$

- Preciznost (*Precision*) je udio točnih procjena u skupu pozitivnih procjena:

$$Preciznost = \frac{TP}{TP + FP} \tag{A.4}$$

- Odziv (*Recall*) je udio točnih procjena u skupu pozitivnih stanja:

$$Odziv = \frac{TP}{TP + FN} \tag{A.5}$$

- Specifičnost (*Specificity*) je udio točnih procjena u skupu negativnih stanja:

$$Specifičnost = \frac{TN}{TN + FP} \tag{A.6}$$

S obzirom da se gore navedene mjere ne mogu promatrati pojedinačno, definira se mjera F kao harmonijska sredina preciznosti i odziva:

$$F = \frac{2 \cdot Preciznost \cdot Odziv}{Preciznost + Odziv} \tag{A.7}$$

Dodatak B: Načini ispitivanja

Za testiranje algoritama se koriste četiri baze slika:

- *Baza1*: 15 (vlastitih) slika BMP formata. Slike su veličine 2592×1944 slikovnih elemenata.
- *Baza2*: 15 slika BMP formata preuzetih s Interneta. Slike su različitih veličina.
- *Baza3*: 20 slika BMP formata preuzetih s Interneta. Slike su različitih veličina.
- *Baza4*: 5 slika TIFF formata preuzetih s Interneta. Slike su veličine 512×512 slikovnih elemenata.

Popis slika iz baza *Baza2*, *Baza3* i *Baza4* se mogu pronaći u popisu preuzetih slika korištenih u ispitivanju. U svim ispitivanjima je faktor kvalitete za JPEG slike jednak 75.

Načini ispitivanja algoritma steganalize se razlikuju. U tablici B.1 se može pronaći točan opis načina na koji je testiran pojedini algoritam steganalize: za određeni test je napisana korištena baza i veličina slika. Kod veličina slike – označava da veličina slike nije promijenjena, a ×256 označava da je slika skalirana tako da joj je veća dimenzija jednaka 256 slikovnih elemenata. Tri su vrste testova:

- *Procjena*: procjena duljine skrivene poruke. Testiranje se provodilo nad porukama relativne veličine 25%, 50% i 75%. Mjere uspješnosti su relativna i apsolutna razlika.
- *Detekcija*: uspješnost detekcije skrivane poruke. U pola slika je skrivena poruka veličine 10%, a u pola slika je skrivena prazna poruka, tj. poruka veličine 0%. Mjere uspješnosti su točnost, preciznost, odziv, specifičnost i F mjera.
- *Prag*: određivanje praga. Testiranje se provodilo nad porukama relativne veličine 25%, 50% i 75%. Kod proširenog hi kvadrat testa se prag mijenjao od 0.1 do 1 s korakom 0.1. Kod F3_stegan algoritma se prag mijenjao od 0.1 do 0.4 s korakom 0.05.

Tablica B.1 Načini ispitivanja

Algoritam	**Procjena**	**Detekcija**	**Prag**
PoV	*Baza1*, 2592×1944	*Baza3*, -	-
RS	*Baza1*, ×256	*Baza3*, ×256	-
dctChi2	*Baza2*, 256×256	*Baza3*, 256×256	-
dctChi2ext	*Baza2*, 256×256	*Baza3*, 256×256	*Baza4*, 256×256
F3_stegan	*Baza2*, 256×256	*Baza3*, 256×256	*Baza4*, 256×256
OG_stegan	*Baza2*, 256×256	*Baza3*, 256×256	-
F5_stegan	*Baza2*, 256×256	*Baza3*, 256×256	-

Kod određivanja kapaciteta steganografskih algoritama koristila se *Baza4* sa slikama veličine 512×512 slikovnih elemenata. Za pronalaženje faktora u formulama za kapacitete kod F3 i F4 algoritma koristile su se slike iz baze *Baza4* veličine 256×256 slikovnih elemenata.

Dodatak C: Pregled implementiranih algoritama

U tablici C.1 se nalazi pregled algoritama implementiranih u sklopu rada. Za svaki steganografski algoritam su dani algoritmi steganalize, kapacitet te podržani formati ulaznih i izlaznih slika. Algoritmi steganalize su obojani ovisno o tome koliko uspješno napadaju promatrani algoritam:

- Zelena: algoritam uspješno napada steganografsku metodu.
- Crvena: algoritam nije uspješan u napadu na steganografsku metodu.
- Plava: algoritam se može prilagoditi tako da uspješno napada steganografsku metodu.

Tablica C.1 Pregled implementiranih algoritama

Algoritam	**Analiza**	*Kapacitet*	*Format (ulaz)*	*Format (izlaz)*
sLSB_slijedno	***PoV*, *RS***	100%	BMP, GIF, TIFF, PNG	BMP, GIF, TIFF, PNG
sLSB_ rand	***PoV*, *RS***	100%	BMP, GIF, TIFF, PNG	BMP, GIF, TIFF, PNG
JSTEG	***dctChi2*, *dctChi2ext*, *F3_stegan*, *OG_stegan***	12.34%	BMP, GIF, TIFF, PNG, JPEG	JPEG
JSTEG_ext	***dctChi2*, *dctChi2ext*, *F3_stegan*, *OG_stegan***	12.34%	BMP, GIF, TIFF, PNG, JPEG	JPEG
OutGuess	***dctChi2*, *dctChi2ext*, *F3_stegan*, *OG_stegan***	6.17%	BMP, GIF, TIFF, PNG, JPEG	JPEG
F3	***dctChi2*, *dctChi2ext*, *F3_stegan*, *F5_stegan*, *OG_stegan***	9.63%	BMP, GIF, TIFF, PNG, JPEG	JPEG
F4	***dctChi2*, *dctChi2ext*, *F3_stegan*, *F5_stegan*, *OG_stegan***	11.86%	BMP, GIF, TIFF, PNG, JPEG	JPEG
F5	***dctChi2*, *dctChi2ext*, *F3_stegan*, *F5_stegan*, *OG_stegan***	12.22%	BMP, GIF, TIFF, PNG, JPEG	JPEG

Literatura

[1] Katzenbeisser S., Petitcolas F.: *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech house, Boston, 2000.

[2] Kipper G., *Investigator's Guide to Steganography*, Auerbach Publications, London, 2004.

[3] CARNet CERT, *Steganografija*, CCERT-PUBDOC-2006-04-154, Revizija v1.0, 2006., Dostupno na: http://sigurnost.lss.hr/documents/LinkedDocuments/CCERT-PUBDOC-2006-04-154.pdf, Pristupano: Travanj 2010.

[4] -, Wikipedia: *Steganography*, *Steganalysis*, *Digital watermarking*, *RC4*, *JPEG*, Dostupno na: http://wikipedia.org/, Pristupano: Svibanj 2010.

[5] Chandramouli, R. Subbalakshmi, K., *Current trends in steganalysis: a critical survey*, Control, Automation, Robotics and Vision Conferenc, 2004. ICARCV 2004, Svezak 2, Str. 964-967, Dostupno na: http://www.ece.stevens-tech.edu/~mouli/iccarv.pdf, Pristupano: Svibanj 2010.

[6] Fridrich J. Goljan M., *Practical Steganalysis of Digital Images – State of the Art*, Proc. SPIE, Svezak 4675, Str. 1-13, Security and Watermarking of Multimedia Contents IV, Dostupno na: http://www.ws.binghamton.edu/fridrich/Research/steganalysis01.pdf, Pristupano: Travanj 2010.

[7] Hrg D., *RC4*, Seminar iz predmeta *Operacijski sustavi 2*, Sveučilište u Zagrebu: Fakultet elektrotehnike i računarstva, 2002., Dostupno na: http://os2.zemris.fer.hr/algoritmi/simetricni/2002_hrg/rc4/index.html, Pristupano: Svibanj 2010.

[8] Zeljković S., *Steganografija*, Hrvatski matematički elektronski časopis *math.e*, Broj 5, Lipanj 2005., Dostupno na: http://e.math.unizg.hr/old/stegano/index.html, Pristupano: Travanj 2010.

[9] Stanley C., *Pairs of Values and the Chi-squared Attack*, Dostupno na: http://orion.math.iastate.edu/dept/thesisarchive/MSCC/CStanleyMSSS05.pdf, Pristupano: Ožujak 2010.

[10] Dalbelo-Bašić B., *Strojno učenje: Nadzirano učenje – evaluacija*, Predavanja iz predmeta *Strojno učenje*, Sveučilište u Zagrebu: Fakultet elektrotehnike i računarstva, Ak. God. 2009./2010.

[11] Fridrich J., Goljan M., Du R., *Reliable Detection of LSB Steganography in Color and Grayscale Images*, IEEE Multimedia, Str. 22-28, 2001., Dostupno na: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.2502, Pristupano: Travanj 2010.

[12] Wang S., Yang B., Niu X., *A Secure Steganography Method based on Genetic Algorithm*, Journal of Information Hiding and Multimedia Signal Processing, Svezak 1, Broj 1, 2010., Dostupno na: http://www.jihmsp.org/~jihmsp/2010/vol1/JIH-MSP-2010-01-004.pdf, Pristupano: Lipanj 2010.

[13] Pandžić I., Bažant A., Ilić Ž., Vrdoljak Z., Kos M., Sinković V., *Uvod u teoriju informacija i kodiranje*, Udžbenik sveučilišta u Zagrebu, Element, Zagreb, 2007.

[14] ITU, *Information technology -- Digital compression and coding of continuous-tone still images: Requirements and guidelines*, Recommendation T.81, 1993., Dostupno na: http://www.w3.org/Graphics/JPEG/itu-t81.pdf, Pristupano: Travanj 2010.

[15] Provos N., Honeyman P., *Hide and Seek: An Introduction to Steganography*, IEEE Security & Privacy, Svezak 1, Broj 3, Str. 32-44, Dostupno na:

http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.3.3820, Pristupano: Travanj 2010.
[16] Provos N., Defending Against Statistical Steganalysis, 10th USENIX Security Symposium, Str. 323-335, 2001., Dostupno na: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.8933, Pristupano: Travanj 2010.
[17] Fridrich J., Goljan M., Hogea D., Attacking the OutGuess, 2002., Dostuno na: http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.18.8955, Pristupano: Travanj 2010.
[18] Westfeld A., *4th International Workshop on Information Hiding*, Str. 289-302, 2001., Dostupno na: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.115.3651, Pristupano: Svibanj 2010.
[19] Fridrich J., Goljan M., Hogea D., *Steganalysis of JPEG Images: Breaking the F5 Algorithm*, 5th International Workshop on Information Hiding, Str. 310-323, 2002., Dostupno na: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.141.1845, Pristupano: Svibanj 2010.
[20] Elezović N., *Vjerojatnost i statistika: Diskretna vjerojatnost*, 0. izdanje, Element, Zagreb, 2001.
[21] Sallee P., *Matlab JPEG Toolbox*, Dostupno na: http://www.philsallee.com/jpegtbx/index.html, Pristupano: Travanj 2010.

Popis preuzetih slika korištenih u ispitivanju

Baza2 je izgrađena od sljedećih petnaest slika:

- batman.bmp: http://2.bp.blogspot.com/_63H-h5Vp7cw/SIARqTz-GmI/AAAAAAAABL8/-6DQgzgHT3A/s320/batman.bmp
- beyonce.bmp: http://cozywallet.com/wp-content/uploads/2009/05/beyonce-makeup.bmp
- charli.bmp: http://kpitalrisk.free.fr/images/stars/0/s_charlize%20theron%207.bmp
- einstein.bmp: http://blog.networksinmotion.com/wp-content/uploads/2009/08/einstein.bmp
- federer.bmp: http://www.robinsapplebyandtaub.com/bridge/beat/wp-content/uploads/2009/06/federer.bmp
- heston.bmp: http://2.bp.blogspot.com/_uKUJj9VMJpA/R_oG245uUJI/AAAAAAAABfY/ndIld3VQa0s/s400/Heston.bmp
- lena.bmp: http://www.indinf.pub.ro/catalinp/proiect/lena512.bmp
- megan.bmp: http://api.ning.com/files/Rh6JgLsX9n-j4zdsIx***YD1FMh8Kk0CD8ynIQm5uPqkGkmIZXYtQ5PoHQbqVrVIqxrnUpWdYxd1J0jFveLHOlf3w5mRwnt-/megan.bmp
- messi.bmp: http://www.dailysoccerblog.net/wp-content/uploads/messi.bmp
- mirko.bmp: http://www.radiosibenik.com/upload/Alilovic.bmp
- pavarotti.bmp: http://money.unblog.fr/files/2007/09/lucianopavarotti.bmp
- shakira.bmp: http://4.bp.blogspot.com/_DLMbxIoKN3c/SAzVpIch-xI/AAAAAAAAD9g/66HwbnoY4L4/s400/untitled.bmp
- sharapova.bmp: http://www.collectiblelegends.com/images/photos/sharapova.bmp
- superman.bmp: http://www.insidesocal.com/outinhollywood/brandon2.bmp
- vader.bmp: http://www.ash-official.com/userfiles/image/Kenami%20Studio%20Pics/Darth_Vader.bmp

Baza3 je izgrađena od sljedećih dvadeset slika:

- auto_1.bmp: http://www.dqfltd.com/images/Classic%20Car%20Boy.bmp
- auto_2.bmp: http://www.freewebs.com/jidousha/car%204.bmp
- avion_1.bmp: http://www.fspilotshop.com/images/prespitbig.bmp
- avion_2.bmp: http://www.fspilotshop.com/images/3.bmp
- brod_1.bmp: http://www.whatsupjacksonville.com/wp-content/uploads/2009/05/spirit-of-massacusetts-tall-ship.bmp
- brod_2.bmp: http://www.bluebookofboats.com/blog/wp-content/uploads/2009/10/brigantine-ship-on-bora-bora.bmp
- crkva_1.bmp: http://www.bellowsfalls.org/art_culture_logos/sca.bmp
- crkva_2.bmp: http://www.fmcreedley.org/:/Pictures/mennonite%20church.bmp
- kuca_1.bmp: http://www.prestige-tours.net/ProductImages/PRESTIGE%20Apt%20Mate%20studio1%20%285%29.bmp
- kuca_2.bmp: http://dallasdirt.dmagazine.com/wp-content/uploads/2008/08/tony-romo-house.bmp
- macka_1.bmp: http://www.layoutlocator.com/graphics/dldimg/532d588c891ca24fc9a664bc46d1a0ce_cat-849.bmp
- macka_2.bmp: http://www.layoutlocator.com/graphics/dldimg/4c1af95fdc23a6aaedd370a1b3eb3686_cat_and_dog-851.bmp
- majmun_1.bmp: http://mimg.ugo.com/201001/35103/cuts/monkey-guy_288x288.bmp
- majmun_2.bmp: http://www.layoutlocator.com/graphics/dldimg/0b6a0ca40b707f188e7811f541ea3259_monkeys_hair-854.bmp
- plaza_1.bmp: http://www.leeabbamonte.com/wp-content/uploads/2009/10/kuta-beach.bmp
- plaza_2.bmp: http://www.leeabbamonte.com/wp-content/uploads/2010/01/como-beach-2.bmp
- stablo_1.bmp: http://www.sfgate.com/blogs/images/sfgate/parenting/2008/03/12/tree340x311.bmp
- stablo_2.bmp: http://www.goodlife.com.ng/uploads/Yetunde-Farinloye_77_palm-tree-3.bmp
- zemlja_1.bmp: http://rockinmama.net/wp-content/uploads/2009/04/earth-day.bmp

- zemlja_2.bmp: http://www.webelowwear.com/wwblog/wp-content/earth.bmp

Baza4 je izgrađena od sljedećih pet slika:

- lady.tiff (4.2.02.tiff): http://sipi.usc.edu/database/database.cgi?volume=misc
- baboon.tiff (4.2.03.tiff): http://sipi.usc.edu/database/database.cgi?volume=misc
- f16.tiff (4.2.05.tiff): http://sipi.usc.edu/database/database.cgi?volume=misc
- lake.tiff (4.2.06.tiff): http://sipi.usc.edu/database/database.cgi?volume=misc
- peppers.tiff (4.2.07.tiff): http://sipi.usc.edu/database/database.cgi?volume=misc