SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Marko Stanec

SKRIVANJE INFORMACIJA U OBJEKT NOSITELJ KORIŠTENJEM LSB ALGORITMA

DIPLOMSKI RAD

Zagreb, 2016.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
POVJERENSTVO ZA DIPLOMSKI ISPIT

Zagreb, 19. travnja 2016.

Zavod: **Zavod za informacijsko komunikacijski promet**
Predmet: **Sigurnost i zaštita informacijsko komunikacijskog sustava**

DIPLOMSKI ZADATAK br. 3495

Pristupnik: **Marko Stanec (0036415867)**
Studij: Promet
Smjer: Informacijsko-komunikacijski promet

Zadatak: **Skrivanje informacija u objekt nositelj korištenjem LSB algoritma**

Opis zadatka:

Potrebno je prepoznati mogućnosti skrivanje informacija u objekt nositelj korištenjem LSB algoritma. Razviti i primjeniti alata za skrivanje/izdvajanje poruke u objekt nositelj korištenjem LSB metode. Ocjeniti prednosti i mane primjene LSB algoritma.

Zadatak uručen pristupniku: 4. ožujka 2016.

Mentor:

izv. prof. dr. sc. Dragan Peraković

Predsjednik povjerenstva za diplomski ispit:

Sveučilište u Zagrebu
Fakultet prometnih znanosti

DIPLOMSKI RAD

SKRIVANJE INFORMACIJA U OBJEKT NOSITELJ KORIŠTENJEM LSB ALGORITMA

HIDING DATA IN CARRIER OBJECT USING LSB ALGORITHM

Mentor: prof. dr. sc. Dragan Peraković

Student: Marko Stanec
JMBAG: 0036415867

Zagreb, rujan 2016.

SKRIVANJE INFORMACIJA U OBJEKT NOSITELJ KORIŠTENJEM LSB ALGORITMA

SAŽETAK

Steganografija je znanstvena disciplina koja proučava metode skrivanja informacija u naizgled bezazlene objekte (npr. slika, audio ili video zapis). Steganografskim postupcima osigurava se tajnost podataka u informacijskom sustavu jer osoba kojoj poruka nije namijenjena nije ni svjesna postojanja iste. Zbog toga je najviše zanimljiva za primjenu u vojne svrhe, odnosno u radu tajnih i obavještajnih službi, gdje je potreban visoki stupanj tajnosti. Rad daje uvid u steganografsku tehniku zamjene bitova najmanje važnosti te ispituje njenu učinkovitost.

KLJUČNE RIJEČI: steganografija, kriptografija, steganaliza, algoritam zamjene bita najmanje važnosti, skrivanje informacija u slikovnoj datoteci, Vigenèrova šifra, generator pseudo-slučajnih brojeva

HIDING DATA IN CARRIER OBJECT USING LSB ALGORITHM

SUMMARY

Steganography is the scientific discipline that studies the methods of hiding informations in seemingly innocuous objects (eg, images, audio or video files). Steganographic procedures ensures the confidentiality of data in the information system because the person to whom the message is not intended is not even aware of its existence. Therefore, it is most interesting for use in military purposes, or in the work of secret and intelligence services, where a high level of secrecy is required. The paper provides insight into steganography technique of replacing bits of least importance, and examines its effectiveness.

KEY WORDS: steganography, cryptography, steganalysis, least significant bit algorithm, hiding informations in image file, Vigenère cipher, pseudo-random number generator

SADRŽAJ

1. UVOD

Jedan od najskupljih i najvažnijih resursa u današnje vrijeme je informacija. Iz tog razloga informaciju je potrebno prikladno zaštititi, no taj problem iz dana u dan postaje sve složeniji. Osnovno pravilo informacijske sigurnosti je da se informacija osigura od neovlaštenih izmjena, da se osigura njena tajnost te da se osigura njena dostupnost.

Tajnost informacija može se ostvariti šifriranjem. Takav oblik zaštite informaciju čini nečitljivom za napadača, no napadač je njenog postojanja i te kako svjestan. Drugi oblik zaštite je da se informacija sakrije u objekt koji sam po sebi već ima određenu funkciju, odnosno već nosi određenu informaciju, npr. slikovna datoteka, audio ili video zapis. U ovom slučaju napadač nije svjestan postojanja tajne poruke. Ovakav oblik zaštite omogućuje steganografija, znanstvena disciplina koja se bavi skrivanjem informacija u naizgled bezazlene objekte.

Tema ovog diplomskog rada je **Skrivanje informacija u objekt nositelj korištenjem LSB algoritma**. Cilj diplomskog rada je razviti alat koji će primjenom LSB algoritma omogućiti skrivanje i izvlačenje poruke iz objekta nositelja. Kao objekt nositelj odabrana je slikovna datoteka s kompresijom bez gubitaka.

Rad je razrađen kroz osam poglavlja. U sljedećem, drugom, poglavlju ukratko su objašnjeni osnovni pojmovi kriptografije. Definiran je osnovni kriptografski sustav te simetrični i asimetrični kriptografski sustavi.

U trećem poglavlju definiran je pojam steganografije te su objašnjena osnovna načela. Definiran je čisti steganografski sustav te steganografski sustav s tajnim i javnim ključem. Također je obrađena sigurnost steganografskog sustava i tipovi steganografije.

Steganaliza je tema četvrtog poglavlja i bavi se ciljevima, metodama te osnovnim tehnikama steganalize, odnosno načinima otkrivanja skrivenih poruka.

U petom poglavlju nalazi se sažeti pregled steganografskih tehnika prostorne i frekvencijske domene koje se koriste kod skrivanja informacija u slikovne datoteke. Zbog lakšeg razumijevanja spomenutih tehnika prvo je definirana slikovna datoteka te metode kompresije s gubicima i bez gubitaka.

Razvoj algoritama implementiranih u razvijeni stego alat izneseni su u šestom poglavlju. Opisan je princip rada jednostavnog, poboljšanog i nasumičnog LSB algoritma te je svaki prikazan pseudo kodom.

U sedmom poglavlju pokazano je korištenje alata i analiza razvijenih algoritama. Ovdje je opisan postupak skrivanja i izvlačenja informacija iz slikovne datoteke korištenjem razvijenog alata te otpornost na vizualne i statističke napade.

Osmo, posljednje numerirano poglavlje je zaključak u kojem se nalaze komentari na razvijene algoritme i alat, prijedlozi za daljnji razvoj njegovih funkcionalnosti te moguća implementacija dodatnih algoritama.

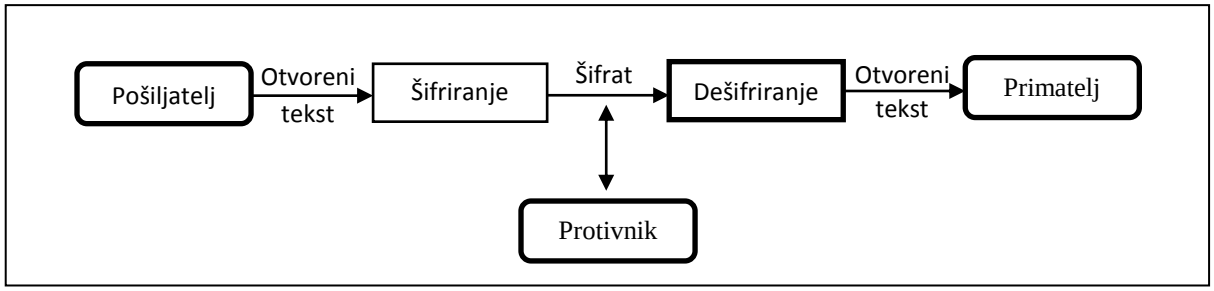
2. KRIPTOGRAFIJA

2.1. Definicija

Kriptografija je znanstvena disciplina koja proučava metode za rješavanje problema sigurne komunikacije putem nesigurnog kanala, odnosno u prisutnosti treće osobe. Sama riječ kriptografija dolazi od grčke riječi kriptós (*grč. κρυπτός*) - "skriven, tajan" i gráfo (*grč. γράφω*) - "pisati", što bi se doslovno moglo prevesti i tajnopis. Osnovni zadatak kriptografije je da omogući sigurnu komunikaciju dvaju osoba, pošiljatelja i primatelja, tako da treća osoba, protivnik, koja nadzire komunikacijski kanala ne može razumjeti ili nije ni svjesna njihove komunikacije, [2], [3], [4].

2.2. Osnove kriptografije

Poruku, koja se šalje, nazivamo *otvoreni tekst* (*eng. plaintext*). Proces kojim se poruka transformira nazivamo *šifriranje* (*eng. encryption*), a dobiveni rezultat *šifrat* (*eng. ciphertext*). Proces vraćanja šifrata u otvoreni tekst nazivamo *dešifriranje* (*eng. decryption*). Ovaj proces prikazan je Slika 1, [1].



Slika 1. *Osnovni kriptografski sustav*
Izvor: [1]

Prema [1] šifriranje i dešifriranje poruke izvodi se pomoću matematičkih funkcija koje nazivamo kriptografski algoritam ili šifra. Želimo li funkcije šifriranja i dešifriranja napisati pomoću matematičkog izraza tada ćemo uvesti oznaku P za otvoreni tekst, C za šifrat, te E i D kao oznake za šifriranje i dešifriranje. Sada možemo reći da funkcija šifriranja E preslikava otvoreni tekst P u šifrat C što je izraženo formulom (1):

$$E(P) = C \tag{1}$$

Gdje oznake imaju sljedeće značenje:

- E – funkcija šifriranja
- P – otvoreni tekst
- C – šifrat

Obrnuto vrijedi da funkcija dešifriranja D preslikava šifrat C natrag u otvoreni tekst P što je izraženo formulom (2):

$$D(C) = P \tag{2}$$

Gdje oznake imaju sljedeće značenje:

- D – funkcija dešifriranja
- C – šifrat
- P – otvoreni tekst

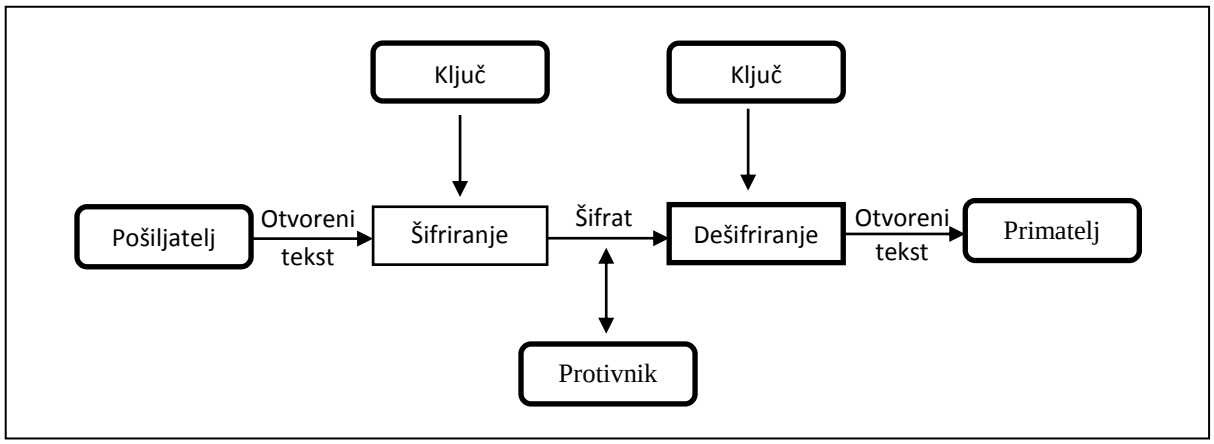
Odnosno dešifriranjem šifriranog otvorenog teksta moramo ponovno dobiti otvoreni tekst, što je izraženo formulom (3):

$$D\big(E(P)\big) = P \tag{3}$$

Ako protivnik presretne šifriranu poruku i zna na koji način funkcija dešifriranja preslikava šifrat u otvoreni tekst, lako može otkriti sadržaj poruke ili istu izmijeniti i takvu je poslati primatelju. Odmah uočavamo da sigurnost ovakvog kriptosustava ovisi isključivo o tome da se način rada algoritma drži tajnim pa govorimo o ograničenom algoritmu. Takav algoritam ne može se koristiti u većim grupama korisnika, jer ako netko napusti grupu ili na bilo koji način ugrozi tajnost rada, svi u grupi primorani su mijenjati algoritam. Nadalje, kod takvog algoritma standardizacija ili kontrola kvalitete su nemogući. Kriptografija nudi rješenje ovog problema uvođenjem ključa (K) koji može imati bilo koju vrijednost. Skup svih mogućih vrijednosti ključeva nazivamo *prostor ključeva*. S obzirom na broj ključeva koji se koriste postoje dvije vrste kriptografskih algoritama: simetrični i asimetrični [1].

2.3. Simetrični kriptografski algoritam

Simetrični algoritam, ponekad nazivan i konvencionalnim algoritmom, je algoritam gdje se ključ za dešifriranje može izračunati poznajući ključ za šifriranje i obrnuto, a najčešće su ti ključevi identični. Sigurnost simetričnog algoritama ovisi o tajnosti ključa pa se takav algoritam još naziva i algoritam s tajnim ključem ili algoritam s jednim ključem, a primjer simetričnog kriptosustava prikazan je Slika 2, [1].



Slika 2. *Simetrični kriptosustav*
Izvor: [4]

Prema [1] pošiljatelj i primatelj moraju unaprijed dogovoriti tajni ključ, a sve dok je vrijednost ključa tajna i komunikacija će se odvijati na siguran način. U ovom slučaju matematički izraz za funkcije šifriranja i dešifriranja koristeći ključ K prikazani su formulama (4) i (5):

$$E_K(P) = C \tag{4}$$

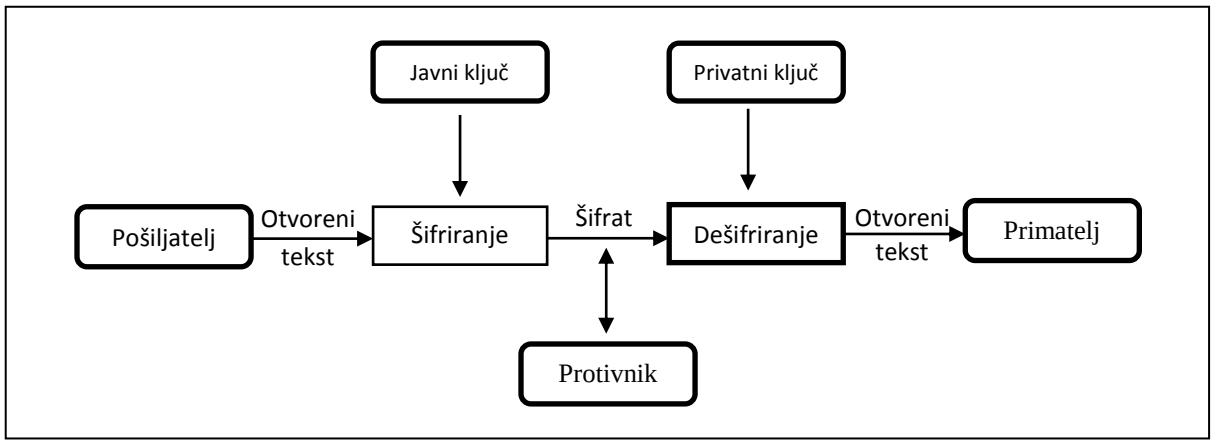
$$D_K(C) = C \tag{5}$$

Gdje oznake imaju sljedeće značenje:

- E_K – funkcija šifriranja ključem K
- D_K – funkcija dešifriranja ključem K
- P – otvoreni tekst
- C – šifrat

2.4. Asimetrični kriptografski algoritam

Asimetrični algoritam, poznat i kao algoritam s javnim ključem, radi na principu dva različita ključa. Ključ za šifriranje razlikuje od ključa za dešifriranje te se iz ključa za šifriranje ne može izračunati ključ za dešifriranje, odnosno to se ne može napraviti u nekom razumnom vremenskom periodu. Ideja ovog algoritma je da ključ za šifriranje bude javno dostupan, tako da bilo tko korištenjem tog ključa može šifrirati poruku, ali samo osoba s odgovarajućim ključem za dešifriranje može otkriti sadržaj šifrirane poruke. Primjer asimetričnog kriptosustava prikazan je na Slika 3, [1].



Slika 3. *Asimetrični kriptosustav*
Izvor: [4]

Ključ za šifriranje, zbog svoje dostupnosti i vidljivosti, naziva se i javni ključ, dok se ključ za dešifriranje naziva privatni ključ. U ovom slučaju matematički izraz za funkcije šifriranja koristeći javni ključ K1 i dešifriranja koristeći privatni ključ K2 izgleda kako je prikazano formulama (6), (7) i (8), [1]:

$$E_{K1}(P) = C \tag{6}$$

$$D_{K2}(C) = P \tag{7}$$

$$D_{K2}(E_{K1}(P)) = P \tag{8}$$

Gdje oznake imaju sljedeće značenje:

- E_{K1} – funkcija šifriranja ključem K1
- D_{K2} – funkcija dešifriranja ključem K2
- P – otvoreni tekst
- C – šifrat

3. STEGANOGRAFIJA

3.1. Definicija

Steganografija je znanstvena disciplina koja proučava metode skrivanja informacija u naizgled bezazleni prijenosni medij s ciljem da nitko ne posumnja u njihovo postojanje. Prijenosni medij naziva se nositelj, a u modernoj steganografiji predstavlja takav skup podataka koji je sastavni dio svakodnevne komunikacije te kao takav ne privlači posebnu pozornost na sebe, npr. tekst, slika, audio ili video zapis. Sama riječ steganografija dolazi od grčke riječi *steganos* (*grč. στεγανός*) što znači "prikriven, zaštićen", i gráfo (*grč. γράφω*) što znači "pisati" te se doslovno može prevesti kao prikriveno pisanje, [5], [6].

Za razliku od kriptografije, steganografija ne mijenja izgled informacije, nego ju samo prikriva i samim time ne privlači pažnju na sebe. Cilj kriptografije je promijeniti izgled informacije do te mjere da je ona nerazumljiva protivniku, ali s druge strane budi sumnju i radoznalost, dok je cilj steganografije informaciju učiniti nevidljivom, [5], [7], [8].

3.2. Osnovna načela steganografije

Prema [7] problem zatvorenika jednostavan je primjer kojim se definiraju osnovna načela steganografije, odnosno steganografskog sustava.

Policija je uhvatila pljačkaša banke i njegovog suučesnika, te ih smjestila u odvojene zatvorske ćelije. Njih dvoje žele dogovoriti plan za bijeg iz zatvora, no komunicirati mogu isključivo preko upravitelja zatvora. Upravitelj zatvora neće dozvoliti šifrirane poruke, i ako primijeti nešto sumnjivo zatvorenici će biti premješteni u samicu. Kako bi dogovorili bijeg zatvorenici moraju slati poruke koje ne privlače pažnju upravitelja, odnosno moraju koristiti tehnike steganografije. Jedan od načina bio bi da međusobno razmjenjuju slike u koje prethodno sakriju informacije o bijegu. Slika plave krave koja pase na zelenom pašnjaku upravitelju bi izgledala potpuno bezazleno i bez znanja o skrivenoj poruci sliku bi predao zatvoreniku kojem je namijenjena, [7].

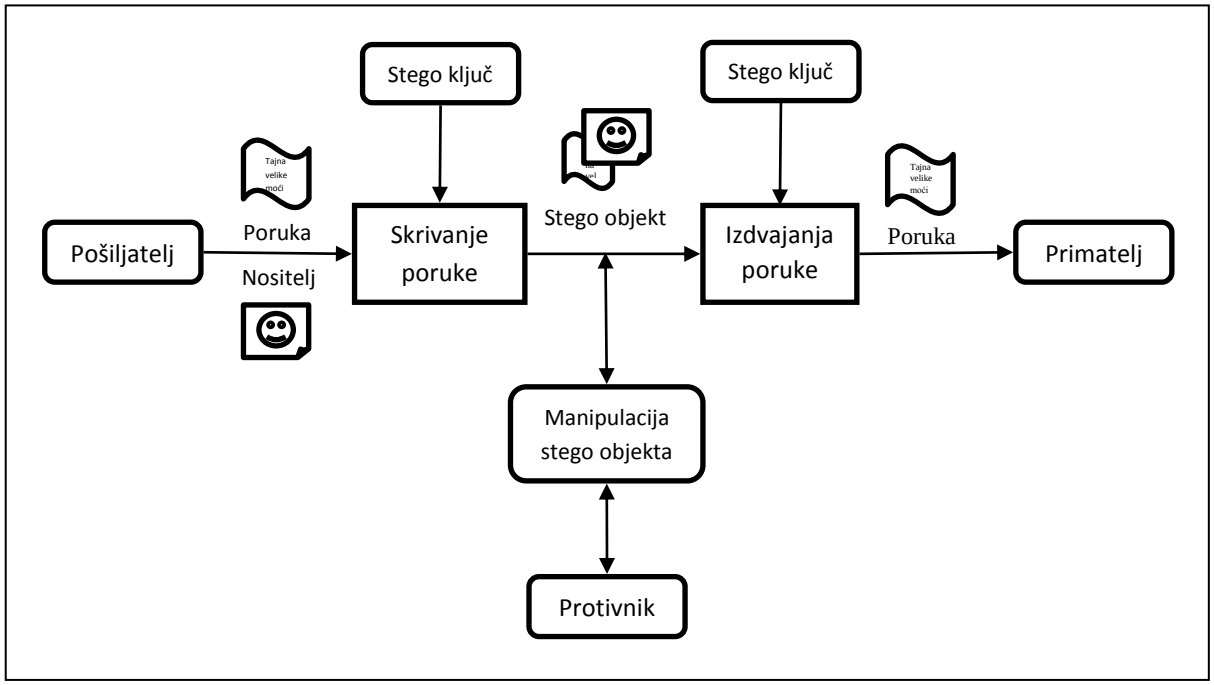
Ako pretpostavimo da je upravitelj mudriji nego što zatvorenici to misle, upravitelj će pokušati otkriti postoji li sumnjivi sadržaj u slikama koje zatvorenici razmjenjuju. Pokušaj otkrivanja sumnjivog sadržaja predstavlja pasivni napad. Prilikom pokušaja otkrivanja skrivene poruke upravitelj može slučajno ili namjerno izmijeniti sliku i tako izmijeniti poruku.

Ako upravitelj namjerno izmjeni sliku kako bi uništio poruku govorimo o aktivnom napadu. No upravitelj može biti još mudriji i u potpunosti izmijeniti sadržaj tajne poruke i sliku predati zatvoreniku kojem je namijenjena. Tako može zavarati zatvorenike koji nisu svjesni izmjena pa u ovom slučaju govorimo o zlonamjernom napadu, [7].

Model problema zatvorenika primjenjiv je na mnoge situacije gdje je moguće primijeniti steganografiju za razmjenu skrivenih poruka. Zatvorenici predstavljaju dvije stane koje žele komunicirati, a upravitelj predstavlja protivnika koji prisluškuje njihovu komunikaciju s ciljem pasivnog, aktivnog ili zlonamjernog napada, [7].

3.3. Steganografski sustav

Rješenje problema zatvorenika je steganografija pa prema tom problemu definiramo model steganografskog sustava koji je prikazan na Slika 4.



Slika 4. *Model steganografskog sustava*

Pošiljatelj postupkom maskiranja poruke i uz pomoć stego ključa umeće poruku u objekt nositelj što za rezultat daje stego objekt. Steganografski objekt ili stego objekt kombinacija je tajne poruke, objekta nositelja i/ili stego ključa. Objekt nositelj (slika, audio ili video zapis) biti će vidljiv svima, i ničime neće izazivati sumnju u postojanje tajne poruke. Korištenjem stego ključa pošiljatelj maskira poruku u objekt nositelj. Istim stego ključem primatelj izdvaja tajnu poruku iz objekta nositelja. Stego ključ može se pojaviti u više oblika, npr. može biti

lozinka ili unaprijed dogovorena pozicija u stego objektu gdje je poruka sakrivena. Protivnik ima mogućnost presretanja i manipulacije stego objekta, a cilj mu je otkrivanje skrivene poruke. Otkrivanjem tajne poruke protivnik je može izdvojiti, izmijeniti ili uništiti, [7].

S obzirom na metode maskiranja i izdvajanja poruke postoje tri vrste steganografskih sustava: čisti steganografski sustav, steganografski sustav s tajnim ključem i steganografski sustav s javnim ključem, [9].

3.3.1. Čisti steganografski sustav

Čisti steganografski sustav uzda se u tajnost i složenost algoritma kojim se poruka maskira unutar objekta nositelja. Sigurnost ovog sustava moguće je povećati kombinirajući tradicionalne kriptografske metode prilikom maskiranja poruke u objekt nositelj. Ako se poruka prije maskiranja šifrira nekim kriptografskim algoritmom, protivnik će teže otkriti da je nešto sakriveno u naizgled bezazlen objekt nositelj. Kod sigurnijih steganografskih sustava ovaj postupak nije potreban, [9].

3.3.2. Steganografski sustav s tajnim ključem

Prema [9] steganografski sustav s tajnim ključem radi na istom principu kao i simetrični kriptografski sustav opisan u prethodnom poglavlju. Prije početka komunikacije pošiljatelj i primatelj sigurnim kanalom razmjene vrijednost tajnog ključa kojim onda maskiraju i izdvajaju poruku iz objekta nositelja. Sigurnost sustava ovisi o čuvanju vrijednosti ključa tajnim.

3.3.3. Steganografski sustav s javnim ključem

Steganografski sustav s javnim ključem počiva na ideji asimetričnog kriptografskog sustava. Pošiljatelju je potreban javni ključ osobe s kojom želi komunicirati kako bi mogao maskirati poruku unutar objekta nositelja, dok samo primatelj sa svojim privatnim ključem može izdvojiti poruku iz stego objekta, [9].

3.4. Sigurnost steganografskog sustava

Probijanje steganografskog sustava sastoji se od tri djela: otkrivanje skrivene poruke, izdvajanje skrivene poruke, te uništavanje, odnosno izmjena skrivene poruke. Pri izradi sigurnog modela steganografskog sustava potrebno je pretpostaviti da protivnik posjeduje neograničenu računalnu snagu, te da je sposoban i željan izvesti razne vrste napada kako bi

kompromitirao sustav. Ako protivnik ne može dokazati postojanje tajne poruke unutar nositelja, tada je sustav teoretski siguran. Ako protivnik postane svjestan postojanje skrivene poruke i može to dokazati sustav je kompromitiran. Protivnik ne treba pročitati skrivenu poruku, dovoljno je da ošteti objekt nositelj, te tako djelomično ili potpuno uništi skrivenu poruku. Zbog toga pri izradi steganografskog sustava posebnu pozornost treba obratiti na aktivne i zlonamjerne napade, [9].

Aktivni napadač koji ne može izdvojiti ili dokazati postojanje tajne poruke može u nositelj dodati slučajni šum, ili dodatno obraditi, odnosno promijeniti format ako se radi o digitalnoj slici i tako omesti tajnu komunikaciju. Zbog toga je potrebno dodatno zaštiti tajnu poruku tako da promjene stego objekta ne utječu na sadržaj poruke, [9].

Što je stego objekt podložniji promjenama to kažemo da je sustav robusniji. No što je sustav robusniji, to je ukupna sigurnost sustava manja, jer se robusnost postiže dodavanjem redundantnih informacija koje uvelike degradiraju objekt nositelj što protivniku olakšava detekciju skrivene poruke. Postoje dva pristupa kako steganografski sustav napraviti robusnijim. Jedan je da se predvide moguće promjene na objektu nositelju, a drugi da se obrnutim postupkom stego objekt vrati u prvobitno stanje nakon napravljene izmjene od strane protivnika, [9].

U prisutnosti zlonamjernog napadača robusnost sustava nije dovoljna. Ako postupak skrivanja poruke ne ovisi o unaprijed dogovorenom tajnom ključu protivnik može krivotvoriti poruke, budući primatelj ne može provjeriti identitet pošiljatelja. Zbog toga steganografski sustav mora biti robustan i siguran. Prema [9] steganografski sustav je siguran ako ispunjava ova četiri uvjeta:

- poruka je skrivena korištenjem javnog algoritma i tajnog ključa koji jedinstveno identificira pošiljatelja
- detektirati, izdvojiti i dokazati postojanje tajne poruke može isključivo osoba koja posjeduje odgovarajući ključ, dok ostali ne bi trebali moći otkriti nikakve tragove o postojanju tajne poruke
- ako protivnik i sazna sadržaj skrivene poruke, ne bi trebala postojati mogućnost za otkrivanje sadržaja ostalih poruka
- detekcija tajne poruke je računski neizvršiva

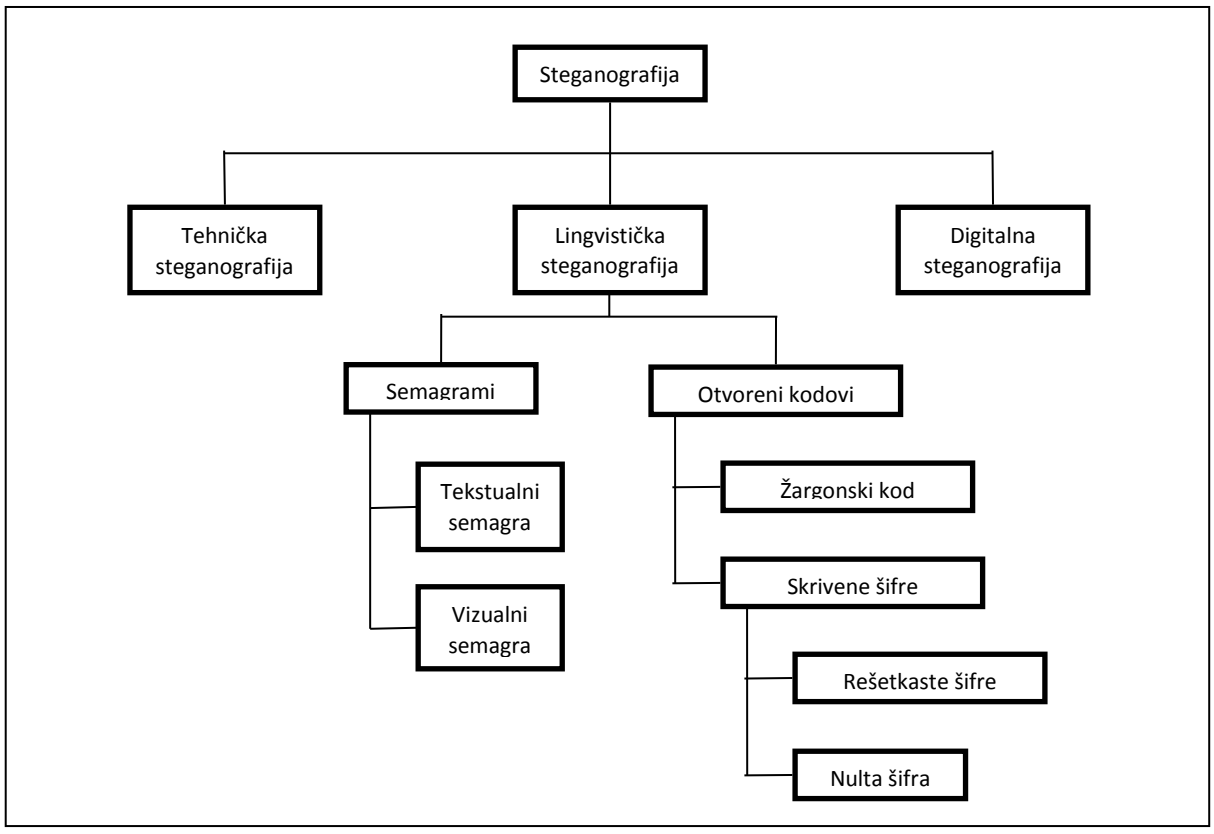
3.5. Tipovi steganografije

Postoje tri osnovna tipa steganografije, a to su tehnička steganografija, lingvistička steganografija i digitalna steganografija, [10].

Tehnička steganografija obuhvaća metode koje tajnu poruku skrivaju korištenjem alata, uređaja ili kemikalija, a u tu skupinu, prema [7], spada:

- Nevidljiva tinta – posebna bezbojna tinta koja ostaje nevidljiva dok ne dođe u doticaj s određenom kemikalijom, toplinom ili posebnim izvorom svjetlosti
- Korištenje skrivanih mjesta – npr. dvostruko dno prtljažnika, peta od cipele, tetovaža ispod kose i sl.
- Mikrofotografija – fotografija veličine i oblika točke, često ne veće od pola milimetra

Lingvistička steganografija obuhvaća metode koje tajnu poruku skrivaju u bezazlen skup informacija. Kao što je prikazano Slika 5, dvije osnovne metode lingvističke steganografije su korištenje semagrama i otvorenih kodova, [10].



Slika 5. *Podjela lingvističke steganografije*
Izvor: [10]

Semagrami skrivaju tajnu poruku uporabom različitih simbola i znakova, a prema [10] dijele se na tekstualne i vizualne semagrame:

- Tekstualni semagram – tajna poruka skriva se određenim promjenama teksta, npr. dodavanja redundantnih praznih mjesta, promjena veličine slova, promjena fonta i sl.
- Vizualni semagram – tajna poruka skriva se korištenjem bezazlenih fizičkih objekata, odnosno njihovim specifičnim razmještajem, npr. raspored objekata na web stranici ili predmeta na stolu

Otvoreni kodovi za skrivanje tajne poruke koriste uobičajene poruke i tekstove iz svakodnevne javne komunikacije. Sadržaj tajne poruke sakriven je na određenim pozicijama u tekstu. Prema [10] otvoreni kodovi dijele se na žargonske kodove ili skrivene šifre.

- Žargonski kod – koristi se jezik koji je razumljiv samo određenoj grupi ljudi
- Skrivene šifre – tajnu poruku moguće je izdvojiti iz stego objekta samo ako se poznaje metoda kojem je poruka umetnuta, a dijele se na rešetkaste šifre i nulte šifre
    - Rešetkaste šifre – preko teksta nosioca postavi se šablona s otvorima, slova koja se pojave na otvorima predstavljaju sadržaj tajne poruke
    - Nulte šifre – skrivena poruka iz teksta se izdvaja primjenom unaprijed dogovorenih pravila, npr. čitanje teksta dijagonalno, čitanje svakog n-tog slova i sl.

Kod digitalne steganografije tajna poruka se skriva u digitalnom mediju, npr. slici, audio ili video zapisu. Prema [10] tehnike digitalne steganografije mogu se podijeliti u osam kategorija:

- Tehnike umetanja – tajna poruka skriva se umetanjem izravno u objekt nositelj (slika, audio ili video zapis) što znatno povećava veličinu objekta pa je lako detektirati postojanje skrivene poruke
- Tehnike izmjene najmanje značajnih bitova – u oktetu bitova mijenja se vrijednost bita s najmanjom vrijednošću jer ima najmanji utjecaj na kvalitetu prijenosnog medija, odnosno teže je detektirati postojanje skrivene poruke
- Tehnike transformacije domene – poruka se umeće u transformacijski prostor koji nastaje prilikom kompresije objekta nositelja. Najčešće se koriste diskretna kosinusna transformacija (DCT), diskretna Fourierova transformacija i diskretna valna transformacija (DWT).

- Tehnike proširenog spektra – u objekt nositelj koji je širokog spektra umeće se tajna poruka koja je uskog spektra. U steganografiji se koriste metode proširenog spektra direktnim slijedom (DSSS), te metode proširenog spektra s frekvencijskim skokom (FHSS)
- Tehnika perceptivnog maskiranja – zbog nemogućnosti ljudskih osjetila vida i sluha da primijete sitne pojedinosti signal tajne poruke postaje neprimjetan protivniku zbog prisutnosti drugih signala
- Tehnika stvaranja nositelja – umjesto odabira objekta nositelja isti se stvara zavisno o tajnoj poruci
- Statističke metode – u svaki signal nositelj umetne se jedan bit tajne poruke i tako stvara statističku promjenu. Promijenjen signal označen je s 1, dok 0 predstavlja nepromijenjen signal.
- Tehnike izobličavanja – izmjenom objekta nositelja skriva se tajna poruka, dok se izdvajanje radi usporedbom izmijenjenog objekta i originala.

4. STEGANALIZA

4.1. Definicija

Steganaliza je znanstvena disciplina i vještina koja se bavi detekcijom stego objekata. Detekcija sumnjivog objekta temelji se na proučavanju varijacija uzoraka bitova i neobično velikih datoteka, [5], [11].

4.2. Ciljevi steganalize

Steganaliza je obrnuti proces u odnosu na steganografiju, i dok je cilj steganografije sakriti tajnu poruku unutar objekta nositelja ciljevi steganalize, prema [5], su:

- identificirati sumnjive skupove podataka (signali ili datoteke) u kojima je potencijalno skrivena tajna poruka
- utvrditi postojanje šuma i suvišnih informacija unutar sumnjive skupine podataka
- utvrditi da li je tajna poruka šifrirana prije umetanja u objekt nositelj
- izdvajanje i dešifriranje tajne poruke iz stego objekta

Kod kriptoanalize predmet promatranja su šifrirani podaci koji očito sadrže poruku, dok je kod steganalize predmet promatranja skup ili nekoliko skupova sumnjivih podataka za koje se ne može biti siguran da sadrže tajnu poruku. Steganalitičar, korištenjem raznih metoda statističke analize, smanjuje skup sumnjivih podataka sve dok ne pronađe steganografski objekt [5], [11].

4.3. Metode steganalize

Steganalitički napadi i analiza skrivenih podataka uključuju različite aktivnosti: detekciju, izdvajanje te onemogućavanje ili uništavanje skrivenih informacija. Prema [5], [9] i [10] vrsta napada ovisi isključivo o informacijama dostupnim steganalitičaru:

- Samo steganografska datoteka (*eng. stegano-only attack*) – dostupna je samo steganografska datoteka nad kojom se potom provode različite analize.
- Poznati nositelj (*eng. known-cover attack*) – dostupni su i steganografska datoteka i objekt nositelj, tj. izvorna datoteka unutar koje je tajna poruka skrivena.
- Poznata poruka (*eng. known-message attack*) – dostupna je tajna poruka i stego objekt.

- Poznati nositelj i odabrana steganografska tehnika (*eng. known-stego attack*) – raspoloživa je steganografska datoteka, steganografski nositelj te steganografski alat, odnosno algoritam korišten za umetanje tajne poruke.
    - Odabrana steganografska tehnika (*eng. chosen-stego attack*) – poznata je i steganografska datoteka i steganografski alat, odnosno algoritam korišten za umetanje tajne poruke.
    - Odabrana poruka (*eng. chosen-message attack*) – poznata poruka i steganografski alat, odnosno algoritam koriste se za kreiranje steganografske datoteke koja se koristi za buduću analizu i usporedbe. Svrha ovog napada je utvrđivanje odgovarajućih uzoraka u steganografskoj datoteci koji mogu ukazati na korištenje određenog steganografskog alata i algoritma.

4.4. Osnovne tehnike steganalize

Skrivanje informacija unutar digitalnog medija uzrokuje izmjene karakteristika tog medija koje se mogu očitovati nekim oblikom degradacije ili neobičnim svojstvima. Slijedi pregled najpopularnijih tehnika steganalize.

4.4.1. Neobični uzorci

Neobični uzorci unutar steganografskih datoteka impliciraju na potencijalno skrivenu poruku unutar istih. Upotrebom različitih alata i tehnika, moguće je identificirati te uzorke. Npr. alatima za analizu diska moguće je filtriranjem pronaći skrivene informacije u nekorištenim particijama. Različiti filtri mogu poslužiti za identificiranje TCP/IP paketa koji sadrže skrivene ili neispravne podatke unutar svog zaglavlja. Pregledom teksta unutar nekog tekstualnog procesora moguće je pronaći male nepravilnosti kod razmještaja riječi i redaka ili suvišne razmake koji impliciraju na postojanje skrivene poruke. Slike mogu sadržavati izobličenja te varijacije u boji i svjetlini boje koje, nakon što se identificiraju nekim alatom, također upućuju na prisustvo skrivenih informacija, [5].

4.4.2. Vizualna detekcija

Analizom ponavljajućih uzoraka moguće je identificirati korišteni steganografski alat ili skrivenu informaciju. Ispitivanje uzoraka provodi se tako da se izvorni steganografski nositelj uspoređuje sa steganografskom datotekom koja sadrži skrivenu poruku. Takav napad naziva se napad s poznatim nositeljem. Usporedbom različitih steganografskih datoteka moguće je

pronaći uzorke koji predstavljaju potpis specifičnog steganografskog alata. Ako izvorni steganografski nositelj nije dostupan, izvedeni potpisi dovoljni su za implikaciju postojanja skrivene poruke te identifikaciju steganografskog alata korištenog za umetanje tajne poruke, [5].

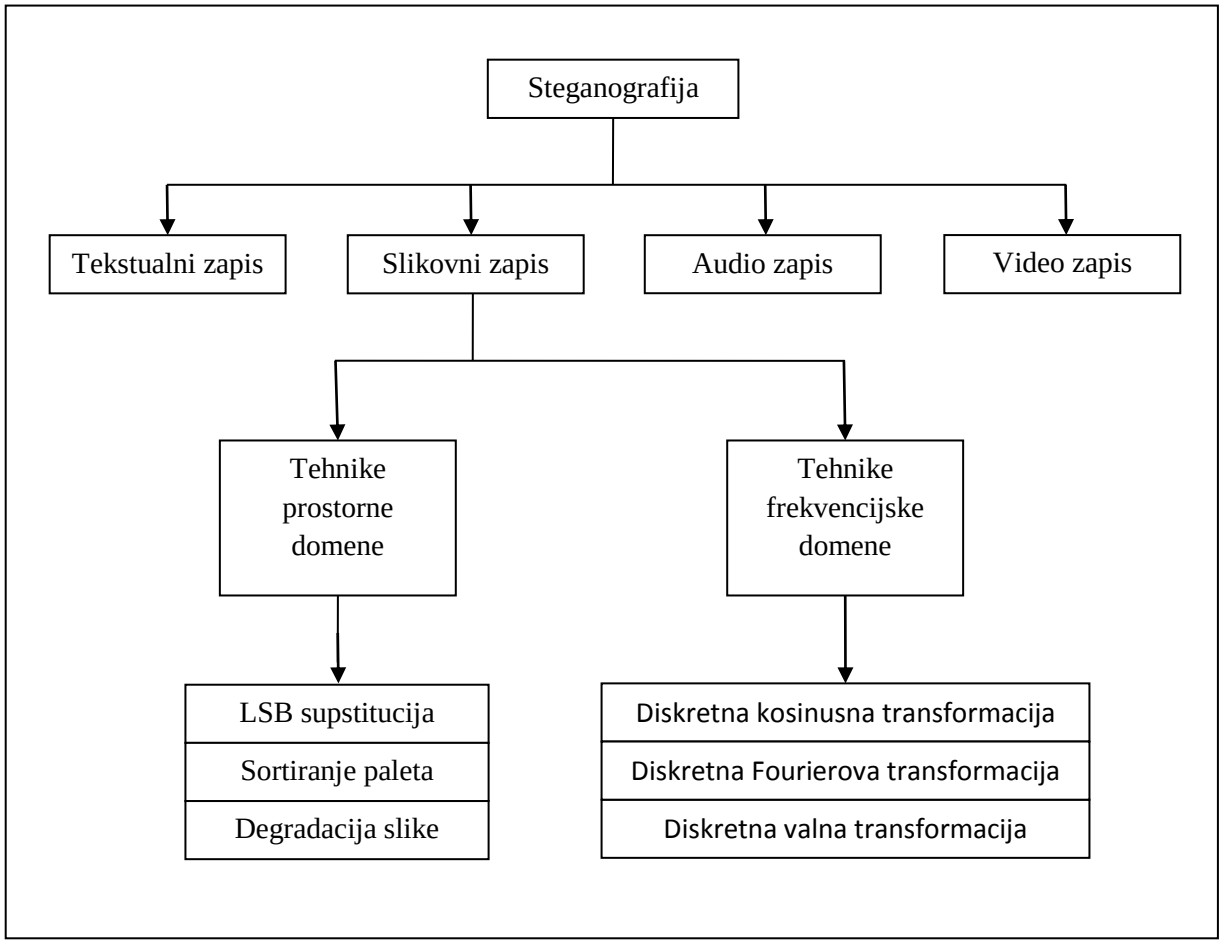
Detekcija takvih potpisa može se automatizirati korištenjem specijalnih alata za detekciju steganografije. Dotični alati obično koriste različite uzorke paleta i potpisa kako bi pronašli piksele koji odstupaju od neke standardne vrijednosti u određenom dijelu slike, [5].

Dodatna indikacija postojanja skrivene informacije unutar slike je njeno nadopunjavanje ili rezanje. Naime, kod nekih steganografskih alata događa se da slika ne odgovara nekoj unaprijed određenoj fiksnoj veličini pa se mora odrezati ili nadopuniti crnim plohama. Nadalje, razlike u veličini između steganografskog nositelja i steganografske datoteke te neobično velik ili malen broj jedinstvenih boja unutra palete slike također upućuju na mogućnost postojanja umetnute poruke u slici, [5].

5. STEGANOGRAFSKE TEHNIKE NAD SLIKOVNIM DATOTEKAMA

Najpopularniji medij za prijenos skrivenih poruka svakako su slike. Steganografija slikovnih datoteka odnosi se na proces skrivanja tajnih, odnosno povjerljivih podataka u sliku. U ovom procesu koriste se postupci pomoću kojih se tajna poruka umetne u slikovnu datoteku i kao takva se šalje primatelju. Primatelj tada obrnutim postupkom izvlači tajnu poruku iz slikovne datoteke, [12].

Postoje brojni steganografski algoritmi pomoću kojih je moguće umetnuti tajnu poruku u objekt nositelj, u ovom slučaju slikovnu datoteku. Prema tehnici koju koriste algoritmi se mogu podijeliti na sljedeći način: tehnike prostorne domene i tehnike frekvencijske domene. Za različite formate slikovnih datoteka koriste se različiti algoritmi. Na Slika 6 prikazana je podjela steganografskih tehnika nad slikovnim datotekama, [12].



Slika 6. *Podjela steganografskih tehnika nad slikovnim datotekama*
Izvor: [12]

5.1. Definicija slikovne datoteke

Računalo sliku vidi kao niz brojeva koji tvore različite intenzitete određenog spektra boje u pojedinim dijelovima slike. Brojčani prikaz formira rešetku, a pojedini dio te rešetke naziva se piksel. Većina slikovnih zapisa sastoji se od pravokutne mape piksela (prikazanih pomoću bitova), njihove pozicije i boje. Pikseli se prikazuju horizontalno, red po red, [13].

Broj bitova koji govori koliko boja je moguće prikazati određuju dubinu boje, odnosno broj bitova potrebnih za predstavljanje boje određenog elementa (piksela) slike. Najmanja dubina boja, odnosno broj bitova za prikaz slike, u trenutnoj kombinaciji boja je 8. To znači da se za opis boje pojedinog piksela koristi 8 bitova. Monokromatske i crnobijele slikovne datoteke koriste 8 bitova za pojedini piksel i tako mogu prikazati 256 različitih boja i nijansi sive, [13].

Digitalne slikovne datoteke u boji obično se pohranjuje u 24-bitne datoteke i koriste RGB model boja, također poznate i kao „prave boje“ (*eng. true color*). Sve kombinacije boja za predstavljanje pojedinog piksela izvedene su iz tri primarne boje: crvene (R), zelene (G) i plave (B), a svaka primarna boja prikazana je pomoću 8 bitova. Tako jedan piksel može prikazati 256 različitih dubina crvene, zelene i plave, što znači da je moguće prikazati 16 777 261 različitih boja. Što je broj boja koje se prikazuje veći, veća je i veličina same datoteke, [13].

5.2. Kompresija slikovne datoteke

Prilikom rada sa slikovnim datotekama većih dubina boja, slikovne datoteke postanu prevelike da bi ih se brzo slalo putem standardne internetske veze. Kako bi se omogućio prijenos i prikaz slikovne datoteke u razumnom vremenu potrebno je primijeniti tehnike za smanjenje veličine slikovnog zapisa. Za spomenute tehnike koriste se matematičke formule pomoću kojih se analiziraju i sažimaju podaci slikovne datoteke, što u konačnici rezultira datotekom manje veličine. Ovaj proces se naziva kompresija (sažimanje) slikovnog zapisa, [13].

Postoje dvije vrste kompresije slikovne datoteke: kompresija s gubitkom (*eng. lossy*) i kompresija bez gubitka (*eng. lossless*). Rezultat obje metode je ušteda prostora za pohranu podataka, ali postupak kompresije se razlikuje. Kompresija s gubitkom kreira datoteke manje veličine tako da odbacuje suvišne podatke iz izvorne slikovne datoteke. Uklanjanje sitnih

detalja, neprimjetnih ljudskom oku, rezultira približnoj aproksimaciji izvorne slikovne datoteke, ali nije točna kopija. Primjer formata slikovne datoteke koja koristi kompresiju s gubitkom je JPEG format, [13].

Kompresija bez gubitka ne uklanja informacije iz izvorne slikovne datoteke, već predstavlja podatke pomoću matematičkih formula. Cjelovitost izvorne slikovne datoteke je očuvana, a podaci dekomprimirane slikovne datoteke identični su podacima izvorne slikovne datoteke. Najpopularniji formati slikovnih datoteka koji koriste kompresiju bez gubitku su GIF i 8 bitni BMP, [13].

Vrsta kompresije slikovne datoteke veoma je bitna prilikom odabira steganografskog algoritma, a odabir kompresije slikovne datoteke, odnosno njenog formata ovisi o potrebama sustava. Prednost i nedostatak oba tipa kompresija je obrnuto proporcionalan te je prikazan Tablica 1. Razvijeni su različiti steganografski algoritmi za oba tipa kompresija slikovnih datoteka te će biti objašnjeni u nastavku, [13].

Tablica 1. *Prednosti i nedostatci kompresije s gubitkom i bez gubitka*

Vrsta kompresije	Prednost	Nedostatak
Kompresija s gubitkom	Manja veličina datoteke	Povećana vjerojatnost djelomičnog ili potpunog gubitka tajne poruke
Kompresija bez gubitka	Očuvanje sadržaja izvorne slikovne datoteke bez mogućnosti gubitka	Veća veličina datoteke

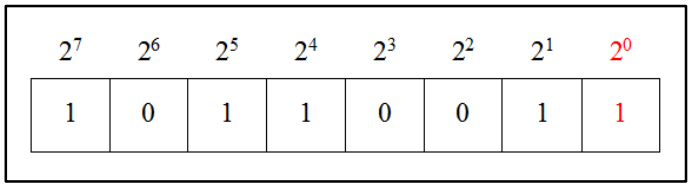
5.3. Tehnike prostorne domene

Tehnike prostorne domene obuhvaćaju metode koje primjenjuju umetanje bitova ili manipulaciju šumovima te su ponekad okarakterizirane kao „jednostavni sustavi“. Formati slikovnih zapisa najpogodniji za steganografske tehnike prostorne domene su formati koji koriste kompresiju bez gubitka, a tehnike ovise o formatu slikovnog zapisa, [13].

5.3.1. Zamjena bita najmanje važnosti

Najčešća steganografska tehnika korištena u radu sa slikovnim datotekama je tehnika zamjene bita najmanje važnosti (*eng. Least Significant Bit Substitution; LSB substitution*).

Pojam „bit najmanje važnosti“ odnosi se na numeričku vrijednost bita u oktetu, odnosno njegovu težinsku vrijednost. Kako je prikazano Slika 7, bit najveće važnosti u oktetu nalazi se na poziciji najveće težinske vrijednosti ($2^7 = 128_{10}$), dok se bit najmanje važnosti nalazi na poziciji najmanje težinske vrijednosti ($2^0 = 1_{10}$) i istaknut je crvenom bojom. Iz tog razloga promjena bita najmanje važnosti ima najmanji učinak na promjenu ukupne vrijednosti okteta, a promjena takvih bitova u svim oktetima koji tvore slikovnu datoteku ima najmanji učinak na promjenu izgleda slikovne datoteke, [5], [13].



Slika 7. *Oktet bitova s prikazom težinskih vrijednosti*

Opisani princip je veoma djelotvoran zbog činjenice da čovjekov optički sustav nije dovoljno osjetljiv da bi primijetio takve promjene u boji. Steganografska tehnika zamjene bita najmanje važnosti zasniva se na rastavljanju tajne poruke na bitove koji se potom pohranjuju na mjesta bitova najmanje važnosti u odabranim oktetima, [5], [13].

U 24 bitnoj slikovnoj datoteci moguće je iskoristiti po jedan bit najmanje važnosti u sve tri boje RGB modela. To znači da u svaki piksel možemo spremiti 3 bita tajne poruke. Za slikovnu datoteku rezolucije 800x600 možemo sakriti 1,440,000 bitova, odnosno 180,000 bajta, [13].

Jednostavnim primjerom prikazat ćemo skrivanje slova M u slikovnu datoteku. Binarna reprezentacija slova M prema proširenom ASCII standardu zauzima 8 bitova, a kako smo ranije objasnili da se u jedan piksel mogu spremiti 3 bita tajne poruke, potrebna su nam 3 piksela za sakrivanje slova M. Uzmimo za primjer tri piksela 24 bitne slikovne datoteke koja izgledaju kao što je prikazano na Slika 8.

```
(00111010   11101001   01001001)
(11000110   01011100   11011001)
(10001010   11001111   01011101)
```

Slika 8. *Prikaz tri piksela u binarnom zapisu*

Binarni zapis slova M je 01001101 i zapisuje se na mjesto najmanje važnih bitova u izvornom skupu okteta. Prikaz slova M umetnutog na pozicije bitova najmanje važnosti prikazano je na Slika 9:

(00111010	*11101001*	*01001000)*
(11000110	*01011101*	*11011001)*
(10001010	*11001111*	*01011101)*

Slika 9. *Slovo M umetnuto u bitove najmanje važnosti*

Iako je bilo potrebno umetnuti 8 bitova, izmijenjena je vrijednost samo 2 podcrtana bita. Za skrivanje poruke maksimalne dozvoljene duljine u prosjeku je potrebno izmijeniti tek polovicu bitova. Dobrim odabirom slike moguće je koristiti i više bitova po oktetu za skrivanje tajne poruke bez narušavanje vizualnog izgleda slikovne datoteke, [13].

U gornjem primjeru bitovi tajne poruke umetani su slijedno od najmanje važnog bita u prvom bajtu pa nadalje. Ako napadač posumnja na postojanje tajne poruke, vrlo lako će otkriti njen sadržaj. Sigurnija inačica bila bi da se bitovi tajne poruke umeću prema unaprijed dogovorenom ključu kojim se određuje redoslijed umetanja bitova tajne poruke. Tako napadač ne može tako lako otkriti sadržaj tajne poruke bez poznavanja vrijednosti tajnog ključa, [13].

U najosnovnijem obliku, za tehniku zamijene bita najmanje važnosti koriste se 24-bitne BMP slike zbog toga što koriste kompresiju bez gubitaka. No taj format nije idealan za skrivanje tajnih poruka jer zahtjeva slike visoke rezolucije. Takve slike mogle bi pobuditi sumnju potencijalnih napadača pa se iz tog razloga tehnika zamijene najmanje važnog bita koristi i za druge formate slikovnih zapisa, [13].

5.3.2. Sortiranje paleta

Tehnika sortiranja paleta koristi se kod slika s paletama boja. Kod slika koje koriste palete boja dovoljno je koristiti samo podskup boja određenog prostora boja za pravilno prikazivanje na slici. Takav format slike sastoji se od dva djela: paleta boja i podataka o slici. Paleta određuje boje listom uređenih parova (i, c_i), gdje je c_i vektor boje pridružen određenom indeksu i, dok se u podacima o slici zapisuje koji indeks se pridružuje kojem pikselu. Ako se slika sastoji od malog broja boja ukupna veličina datoteke se drastično smanjuje, [9].

Postoje dva osnovna načina za skrivanje informacija u slike bazirane na paletama boja, a to su manipulacija paleta ili manipulacija podataka o slici. Mogu se koristiti bitovi najmanje važnosti vektora boja kao što je prethodno objašnjeno, međutim kako palete mogu biti sortirane bilo kojim redoslijedom, informacije se mogu sakriti redoslijedom spremanja boja u

palete. Budući da postoji n! načina sortiranja paleta stvara se dovoljno prostora za umetnuti kratku poruku. Takva metoda nije dovoljno pouzdana, jer napadač može uništiti tajnu poruku premještanjem paleta, a da pritom ne promijeni izgled slike, [9].

Drugi način je da se tajna poruka sakrije u dio koji definira podatke o slici. S obzirom ne to da vrijednosti susjednih paleta boja ne trebaju nužno biti perceptivno slične, tehnika zamjene bitova najmanje važnosti nije primjenjiva. Zbog toga je prvo potrebno sortirati palete prema komponenti koja određuje svjetlinu boje, jer je ljudsko oko osjetljivije na takve promijene. Nakon pravilnog sortiranja paleta može se primijeniti tehnika zamjene bita najmanje važnosti, [9].

Postoji još jedna tehnika koja se razlikuje od prethodnih i ne zahtijeva da palete budu sortirane, a ta je da se za svaki piksel odredi skup najbližih boja računanjem Euklidske udaljenosti u RGB prostoru prema formuli (9):

$$d = \sqrt{R^2 + G^2 + B^2} \tag{9}$$

Gdje oznake imaju sljedeće značenje:

- d – udaljenost piksela od skupa najbližih boja
- R – komponenta piksela za određivanje dubine crvene boje
- G – komponenta piksela za određivanje dubine zelene boje
- B – komponenta piksela za određivanje dubine plave boje

Kada se odredi prva najbliža boja, traži se iduća kojoj se paritet poklapa s bitom tajne poruke koji se umeće. Kada se pronađe takva boja, piksel mijenja boju, [9].

5.3.3. Degradacija slike

Degradacija slike *(eng. image downgrading)* je tehnika kojom možemo sakriti jednu slikovnu datoteku u drugu manipulacijom bitova najmanje važnosti. Ovo je specifičan slučaj jer je slika objekt nositelj i tajna poruka. Za umetanje tajne slike u sliku koja služi kao objekt nositelj potrebno je uzeti slike jednakih dimenzija, kao što je prikazano Slika 10 te nakon toga izmijeniti četiri bita najmanje važnosti slike koja služi kao nositelj i na to mjesto zapisati četiri bita najveće važnosti tajne slike. Izvlačenje tajne slike iz stego objekta izvodi se tako da se pročitaju četiri bita najmanje važnosti svakog elementa stego objekta te se nadopune s četiri proizvoljna bita, npr. nulama. Tako dobije se aproksimacija tajne slike zadovoljavajuće

kvalitete. Slika 11 prikazuje stego sliku i izvučenu tajnu sliku koja je skrivena prethodno opisanom tehnikom zamjene četiri bita najmanje važnosti, [9], [14], [15].

Slika 10. *Slika objekt nositelj i tajna slika*
Izvor: [15]

Slika 11. *Stego slika s umetnutom tajnom slikom i izvučena tajna slika*
Izvor: [15]

5.4. Tehnike frekvencijske domene

Tehnike frekvencijske domene uključuju metode manipulacije algoritama i transformacije slikovnih zapisa. Tim metodama tajne poruke se skrivaju na značajnijim mjestima slikovnog zapisa, što ih čini otpornijim na detekciju i izmjene. Mnoge metode frekvencijske domene neovisne su o formatu slikovnog zapisa pa tajna poruka ostaje nepromijenjena bez obzira na promjenu između kompresije s gubitkom ili bez gubitka, [13].

Kako bi razumjeli tehnike frekvencijske domene potrebno je objasniti format slikovnog zapisa koji se najčešće povezuje s ovim tehnikama, a to je JPEG. Kako bi komprimirali slikovni zapis u JPEG format prvi korak je konvertiranje RBG modela u YUV model, gdje Y komponenta predstavlja svjetlinu (*eng. luminance*), a komponente U i V predstavljaju boju

(*eng. chrominance*). JPEG kompresija zasniva se na činjenici da je ljudsko oko osjetljivije na promjenu svjetline u odnosu na promjenu boje piksela pa poduzorkovanjem (*eng. downsampling*) komponenata boje smanjuje ukupnu veličinu slikovnog zapisa, [13].

Sljedeći korak je transformacija slikovnog zapisa korištenjem jedne od tehnika transformacije domene (DCT, DFT, DWT) koje će biti detaljnije objašnjene u nastavku. Nakon transformacije slikovnog zapisa slijedi kvantizacija. Ovaj proces zasniva se na još jednoj činjenici da ljudsko oko dobro uočava razlike u osvijetljenosti površine, ali teško raspoznaje točan doprinos visokih frekvencija ukupne osvijetljenosti pa je moguće smanjiti količine visokih frekvencija bez prevelikog utjecaja na izgled slikovnog zapisa. To se postiže dijeljenjem svih vrijednosti pojedinog bloka kvantizacijskim faktorom i zaokruživanjem na cjelobrojnu vrijednost, što uzrokuje gubitak informacije ali i smanjuje količinu podataka. Dodatno smanjivanje postiže se Huffmanovim kodiranjem, [13].

5.4.1. Diskretna kosinusna transformacija

Diskretna kosinusna transformacija pomaže pri dijeljenju slikovnog zapisa u dijelove različite važnosti, uzimajući u obzir vizualnu kvalitetu slikovnog zapisa. Prema [7] ova transformacija zasniva se na dvije tehnike smanjivanja količine informacija potrebnih za prikaz slikovne datoteke:

1. Kvantizacija DCT koeficijenata slikovne datoteke – postupak smanjivanja broja bitova zaokruživanjem na određene (diskretne) vrijednosti.
2. Entropijsko kodiranje kvatiziranih koeficijenata – tehnika što kompaktnijeg prikaza kvatiziranih podataka.

Jednostavan primjer kvantizacije je zaokruživanje realnih brojeva na cjelobrojne vrijednosti. Prikaz realnog broja između 0 i 7 s velikom preciznošću zahtjeva dosta bitova, ali ako tu vrijednost zaokružimo na najbližu cjelobrojnu vrijednost dobijemo kvantizaciju koju je moguće prikazati korištenjem samo tri bita, [7].

Na primjer, 4.65987 zaokruženo na 5 zauzima manje bitova pa tako možemo smanjiti broj mogućih vrijednosti i broja bitova potrebnih da ih se prikaže uz gubitak informacija. „Finijom“ kvantizacijom dobivamo veći broj mogućih vrijednosti i gubimo manje informacija, [7].

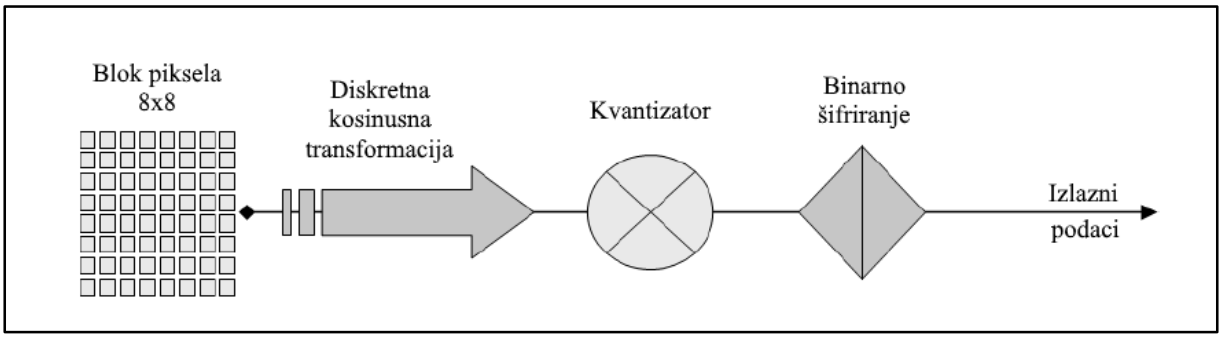
Kod JPEG kompresije svaki koeficijent kosinusne transformacije kvantizira se korištenjem težinskog faktora koji ovisi o frekvencijama pojedinog koeficijenta. Koeficijenti svakog 8x8 bloka piksela podijeljeni su odgovarajućim elementom 8x8 kvantizacijske matrice, a rezultat je zaokružen na najbližu cjelobrojnu vrijednost. Za svaki blok piksela 8x8 koji sadrži vrijednost piksela f(x,y), koeficijenti f(u,v) se određuju prema formuli (10), [12]:

$$F(u,v) = \frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2x+1)v\pi}{16}\right] \tag{10}$$

Gdje je:

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}}, ako\ je\ u \leq 0 \\ 1, ako\ je\ u > 0 \end{cases} \tag{11}$$

Slika 12 prikazan je postupak JPEG kompresije slikovne datoteke korištenjem DCT tehnikom, [7].

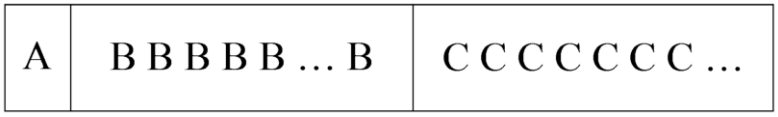


Slika 12. *Postupak JPEG kompresije slikovne datoteke*
Izvor: [7]

1. Podjela slikovnog zapisa u blokove piksela 8x8 i računanje DCT koeficijenata za svaki blok.
2. Diskretna kosinusna transformacija pomaže pri podjeli slikovnog zapisa u dijelove različitih važnosti s obzirom na kvalitetu vizualne percepcije slikovnog zapisa.
3. Kvantizator zaokružuje DCT koeficijente prema kvantizacijskoj matrici. Ubacivanje tajnog sadržaja vrši se nakon ovog koraka. Bitovi najniže vrijednosti svih frekvencijskih koeficijenata većih od nule mijenjaju se bitovima tajnog sadržaja.
4. Huffmannov kôder obavlja proces binarnog šifriranja koje frekvencije boja mijenja u brojčanu vrijednost, prilikom čega dolazi do dodatnih gubitaka, ali također omogućuje veći stupanj kompresije.

5. JPEG kompresija koristi kôd promjenjive dužine kojim zapisuje komprimirani tok podataka u izlaznu datoteku, dok kod dekompresije vraća DCT koeficijentima prvobitnu vrijednost, obavlja inverziju i prikazuje slikovni zapis.

Format zapisa sadržaja tajne poruke koji se umetne na pozicije najmanje važnih bitova izgleda kako je prikazano na Slika 13.



Slika 13. *Format zapisa sadržaja tajne poruke*
Izvor: [7]

- „A“ zauzima 5 bitova i predstavlja duljinu polja „B“ (u bitovima).
- „B“ je određeni broj bitova od 0 do 31 i predstavlja duljinu umetnutog sadržaja (u bajtovima). Raspon vrijednosti za „B“ je od 0 do milijardu.
- „C“ predstavlja bitove umetnutog sadržaja

Ovakav format čini sadržaj tajne poruke neprimjetan koliko je to moguće. Ali neprimjetnost rješenje samo djela problema. Količina prostora za pohranu tajnog sadržaja je zadovoljavajuća, ali nije izvanredna. Testiranja su pokazala da kompresija tajnog sadržaj prije umetanja u slikovni zapis ne radi gotovo nikakvu štetu prilikom kompresije, [7].

5.4.2. Diskretna Fourierova transformacija

Diskretna Fourierova transformacija pretvara slikovni zapis iz prostorne u frekvencijsku domenu tako da ih pretvara u vrijednosti sinusa i kosinusa. Primjenjiva je kod filtriranja i rekonstrukcije slikovnog zapisa, kao i kod kompresije. Fourierova transformacija ne uključuje sve frekvencije iz kojih proizlazi slikovni zapis, već je sačinjena samo od onih uzoraka koji su dovoljni da bi opisali originalni slikovni zapis.

Diskretna Fourierova transformacija za vektor *x* dužine *n* je drugi vektor *y* dužine *n*, što je izraženo formulom (12):

$$y_{p+1} = \sum_{j=0}^{n-1} \omega^{jp} x_{j+1} \tag{12}$$

Gdje je:

$$\omega = e^{-2\pi i/n} \tag{13}$$

5.4.3. Diskretna valna transformacija

Valne (*eng. wavelet*) funkcije vremenski su ograničene i mogu imati bilo kakav oblik. Multirezolucijsko predstavljanje signala je osnovno načelo valne transformacije, koja za razliku od Fourierove transformacije, signal prikazuje istodobno u vremenskoj i frekvencijskoj domeni. Signal se promatra u vremenskim intervalima i za svaki takav interval se računa spektar, [16].

Kod valnih funkcija se ne koristi pojam spektra, već je uveden termin skala. Skala je obrnuto proporcionalna frekvencijskom pojasu. Fourierovom transformacijom signal se predstavlja pomoću sinusnih i kosinusnih funkcija dok se kod valne transformacije prikazuje tzv. valnim funkcijama. Kod valne transformacije, cijela se slika komprimira, tj. ne dijeli se na blokove kao što je slučaj kod diskretne kosinusne transformacije, [16].

Sve valne funkcije generirane su postupkom skaliranja i translacije iz iste funkcije, prikazane formulom (14),koja se zove osnovna ili *„mother wavelet“* funkcija, [16]:

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi\left(\frac{t-\tau}{s}\right) \tag{14}$$

6. RAZVOJ LSB ALGORITAMA ZA SKRIVANJE PORUKE

Zadatak ovog diplomskog rada je skrivanje tajne poruke u objekt nositelj pa je prvi korak u izradi bio odabir objekta nositelja. Za tu potrebu odabrana je slikovna datoteka formata PNG s kompresijom bez gubitka zbog jednostavnosti izvedbe.

Za razvoj stego alata, pomoću kojeg će se obavljati umetanje, odnosno skrivanje tajne poruke u objekt nositelj, odabran je programski jezik Python. Uz standardne Python biblioteke i module korištene su i biblioteke PIL, Crypto i loremipsum. Najbitnija korištena biblioteka je PIL biblioteka (Python Image Library) koja omogućuje obradu slikovnih datoteka raznih formata. Razvijeni alat pokreće se iz komandne linije, odnosno nema grafičko sučelje te je razvijen i testiran na Ubuntu 14.04 LTS operacijskom sustavu, jednom od najpopularnijih Linux distribucija.

Ideja pri izradi alata bila je implementirati jednostavni (osnovni) LSB algoritam te ga nakon toga poboljšati kako bi se povećala ukupna kompleksnost algoritma, a samim time i sigurnost u pohrani i prijenosu informacije. Tako su razvijena tri algoritma: jednostavni LSB algoritam, poboljšani LSB algoritam te nasumični LSB algoritam. Način rada spomenutih algoritama biti će detaljnije opisan u sljedećim poglavljima.

6.1. Jednostavni LSB algoritam

Osnovni način skrivanja poruke u slikovnu datoteku je metoda zamjene bita najmanje važnosti u kojoj se najmanje važni bit RBG komponenti piksela zamjeni bitom tajne poruke.

Algoritam je jednostavan jer ne zahtijeva dodatne manipulacije nad slikovnom datotekom. Zbog toga je cijeli postupak izrade i implementacije algoritma dosta pojednostavljen, a sigurnost ovisi isključivo o algoritmu.

6.1.1. Skrivanje poruke jednostavnim LSB algoritmom

Proces skrivanja poruke korištenjem jednostavnog LSB algoritma prikazan je pseudo kodom na Slika 14 iz čega je vidljivo da se cijeli algoritam može implementirati u nekoliko linija koda. Algoritam radi tako da ugniježđenom petljom dohvati svaki piksel slikovne datoteke, te slijedno u svaku komponentu piksela umetne po jedan bit tajne poruke na mjesto bita najmanje važnosti. Zamjena bitova se obavlja dok se ne umetnu svi bitovi tajne poruke.

```
def jednostavno_lsb_skrivanje(slika, poruka):
    index = 0
    binarni_zapis = pretvori_u_binarni_zapis(poruka)
    za red u slika.visina:
        za kolona u slika.sirina:
            rgb = dohvati_rgb((red, kolona))
            ako je index < duljina(binarni_zapis):
                zamjeni_lsb(rgb,binarni_zapis[index])
                index += 1
    spremi_stego_sliku()
```

Slika 14. *Pseudo kod jednostavnog LSB algoritma za skrivanje poruke*

6.1.2. Izvlačenje poruke jednostavnim LSB algoritmom

Proces izvlačenja poruke korištenjem jednostavnog LSB algoritma još je jednostavniji od skrivanja poruke, a pseudo kod za izvlačenje poruke korištenjem jednostavnog LSB algoritma prikazan je na Slika 15. Budući da se prilikom procesa skrivanja bitovi poruke zapisuju slijedno, poznat nam je redoslijed kojim ćemo izvlačiti podatke iz slike.

```
def jednostavno_lsb_izvlacenje(slika):
    niz_bitova = ''
    za red u slika.visina:
        za kolona u slika.sirina:
            rgb = dohvati_rgb((red, kolona))
            niz_bitova += procitaj_lsb(rgb)
    poruka = pretvori_u_ASCII(niz_bitova)
    vrati poruka
```

Slika 15. *Pseudo kod jednostavnog LSB algoritma za izvlačenje poruke*

Algoritam radi tako da se ugniježđenom petljom dohvatiti svaki piksel slike te iz njegovih RGB komponenti pročita bit najmanje važnosti. Nakon toga se niz bitova čita slijedno u grupama od osam, odnosno jedan bajt, te se pretvara u čitljivi ASCII znak. ASCII znakovi će biti razumljivi i čitljivi do dijela niza bitova na kojem završava tajna poruka, nakon čega ćemo dobiti niz nečitljivih znakova. Iz tog razloga prilikom procesa spremanja tajnoj poruci dodajemo jedinstvenu oznaku za kraj poruke kako bi izvukli samo tajnu poruku bez nepotrebnih i nečitljivih znakova.

6.2. Poboljšani LSB algoritam

Primjenom jednostavnog LSB algoritma možemo lako sakriti tajnu poruku u slikovnu datoteku, međutim napadač koji posumnja u njeno postojanje može lako doći do njenog sadržaja ako pokuša čitati bitove najmanje važnosti RGB komponenti pojedinog piksela.

Iz tog razloga prethodni algoritam ćemo poboljšati kako bi potencijalnog napadača zavarali o postojanju tajne poruke. Za tu svrhu koristiti ćemo kriptografsku metodu šifriranja poznatiju kao Vigenèrova šifra.

6.2.1. Vigenèrova šifra

Vigenèreova šifra jedan je od najpopularnijih kriptosustava u povijesti te pripada *polialfabetskim kriptosustavima*. Ovo je dobro poznata šifra zato što je laka za razumijevanje i primjenu. Ipak početnicima izgleda kao neprobojna pa je zato i dobila epitet - neprobojna šifra (*fr. le chiffre indéchiffrable*), [19].

Vigenèreova šifra je metoda šifriranja abecednog teksta korištenjem serije Cezarovih šifri, zasnovanih na slovima ključa. Kod nje se svako slovo otvorenog teksta može preslikati u jedno od *m* mogućih slova (gdje je *m* duljina ključa), [18], [19].

Jednostavan primjer šifriranja Vigenèreovom šifrom za otvoreni tekst STEGANOGRAFIJA i ključ FPZ izgleda tako da se slova zamjene rednim brojem abecede. Tablica 2 prikazuje slova abecede s pripadajućim rednim brojem. Vrijednosti rednog broja slova se zbroje, te se uzme ostatak dijeljenja s ukupnim brojem slova u abecedi. Dobivena vrijednost predstavlja slovo šifrata.

Tablica 2. *Slova abecede s pripadajućim rednim brojem*

A	B	C	Ć	Č	D	DŽ	Đ	E	F	G	H	I	J	K	L	LJ	M	N	NJ	O	P	R	S	Š	T	U	V	Z	Ž
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Primjer šifriranja prvog slova otvorenog teksta i prvog slova ključa:

$$(S + F) \% 30 \rightarrow (24 + 10) \% 30 = 34 \% 30 = 4 \rightarrow Ć$$

Dobiveni šifrat otvorenog teksta STEGANOGRAFIJA i ključa FPZ prikazan je Slika 16.

Otvoreni tekst:	S	T	E	G	A	N	O	G	R	A	F	I	J	A
Ključ:	F	P	Z	F	P	Z	F	P	Z	F	P	Z	F	P
Šifrat:	Ć	M	Đ	O	R	M	A	C	P	G	B	H	S	R

Slika 16. *Primjer šifriranja Vigenèreovom šifrom*

Iz gornjeg primjera možemo izvesti opći algebarski oblik Vigenèreove šifre za šifriranje i dešifriranje nekog otvorenog teksta *P* ključem *K* iz zadanog skupa znakova duljine *l.* Formulama (15) i (16) prikazane su funkcije Vigenèreove šifre za šifriranje i dešifriranje.

$$C_i = E_K(P_i) = (P_i + K_i) mod\ l \quad (15)$$

$$P_i = D_K(C_i) = (C_i - K_i) mod\ l \quad (16)$$

Gdje je:

- C – šifrat
- P – otvoreni tekst
- K – ključ
- l – duljina zadanog skupa znakova
- E_K – funkcija šifriranja ključem K
- D_K – funkcija dešifriranja ključem K

6.2.2. Skrivanje poruke poboljšanim LSB algoritmom

Poboljšani LSB algoritam u osnovi je jednostavni LSB algoritam, samo što se prije skrivanja poruka šifrira Vigenèrovom šifrom te se u sliku skriva šifrat. U slučaju da napadač posumnja u postojanje tajne poruke te je pokuša izvući jednostavnim LSB algoritmom umjesto poruke će dobiti šifrat koji sadrži niz nerazumljivih i nečitljivih znakova. Na Slika 17 prikazan je pseudo kod algoritma pomoću kojeg se otvoreni tekst šifrira uz pomoć ključa kojeg je unio korisnik.

Ovaj algoritam predstavlja jednostavnu implementaciju Vigenèrovog šifriranja opisnog funkcijom (15) u prethodnom poglavlju. Kako proširena ASCII tablica sadrži 255 znakova, tako duljina zadanog skupa znakova iznosi 255, što znači da dobiveni šifrat može poprimiti vrijednost znakova koji se ne prikazuju na ekranu kao što su na primjer 'NULL' znak, znak za novi red, kontrolni znakovi i slično. Takav izgled šifrata na prvi pogled neće djelovati kao sadržaj skrivene poruke, što će potencijalnom napadaču otežati otkrivanje sadržaja. Nakon šifriranja tajne poruke, primjenjujemo jednostavni LSB algoritam za skrivanje šifrata.

```
def vigenerovo_sifriranje(poruka, kljuc):
    kljuc = prosiri_kljuc_na_duljinu_poruke(kljuc,duljina(poruka))
    za i u rasponu(0, duljina(poruka)):
        sifrat += ((poruka[i] + kljuc[i]) % 255)
    vrati sifrat

def poboljsano_lsb_skrivanje(slika, poruka, kljuc):
    sifrat = vigenerovo_sifriranje(poruka, kljuc)
    jednostavno_lsb_skrivanje(slika,sifrat)
```

Slika 17. *Pseudo kod poboljšanog LSB algoritma za skrivanje poruke*

6.2.3. Izvlačenje poruke poboljšanim LSB algoritmom

Izvlačenje tajne poruke iz slike u koju je ista skrivena poboljšanim LSB algoritmom zahtijeva obrnuti proces uz uvjet poznavanja vrijednosti ključa kojim je obavljeno šifriranje. Bez točne vrijednosti ključa iz slike će se izvući niz nerazumljivih i nečitljivih znakova. Proces izvlačenja sastoji se od dva djela, prvo se izvuku bitovi najmanje važnosti RGB komponenti svih piksela u slici te se pretvore u ASCII znakove, a nakon toga se primjenjuje algoritam Vigenèrovog dešifriranja, čiji je pseudo kod prikazan na Slika 18.

```
def vigenerovo_desifriranje(sifrat, kljuc):
    kljuc = prosiri_kljuc_na_duljinu_sifrata(kljuc,duljina(sifrat))
    za i u rasponu(0, duljina(sifrat)):
        poruka += ((sifrat[i] - kljuc[i]) % 255)
    vrati poruka

def poboljsano_lsb_izvlacenje(slika, kljuc):
    sifrat = jednostavno_lsb_izvlacenje(slika)
    poruka = vigenerovo_desifriranje(sifrata)
    vrati poruka
```

Slika 18. *Pseudo kod poboljšanog LSB algoritma za izvlačenje poruke*

Baš kao i kod algoritma za šifriranje otvorenog teksta ponovno se primjenjuje opći zapis Vigenèrove šifre za dešifriranje šifrata, izražene formulom (16), na način da se svakom znaku šifrata oduzme znak ključa te se nakon toga uzima vrijednost ostatka dijeljenja brojem 255, što daje vrijednost znaka otvorenog teksta. Kao i kod jednostavnog LSB algoritma znakovi se čitaju do jedinstvene oznake za kraj poruke.

6.3. Nasumični LSB algoritam

Poboljšanim LSB algoritmom omogućena je teža detekcija skrivene poruke u slučaju da napadač posumnja u njeno postojanje i pokuša je pročitati jednostavnim izvlačenjem bitova najmanje važnosti. No ako je napadač dovoljno uporan i posumnja da je skrivena poruka zapravo šifrat, samo je pitanje vremena kada će zaključiti da se radi o Vigenèrovoj šifra, jer prema Parkeru Hittu *„Uspjeh u rješavanju nepoznatih šifara mjeri se ovim četirima pokazateljima, redom kako su ovdje navedeni: upornošću, pažljivim postupcima analize, intuicijom i srećom“*. Iako Vigenèrova šifra na prvi pogled izgleda neprobojna, kriptoanalitičari su otkrili metode pomoću kojih mogu odgonetnuti ključ šifriranja, a samim time mogu iz šifrata dobiti otvoreni tekst.

Iz gore navedenog razloga potrebna je još jedna nadogradnja LSB algoritma kako bi se dobila veća kompleksnost i omogućilo gotovo nemoguće izvlačenje tajne poruke iz slikovne datoteke. Kako bi se to ostvarilo primijeniti će se generator pseudo-slučajnih brojeva za nasumičan odabir piksela u koje će se spremati bitovi tajne poruke.

6.3.1. Generator pseudo-slučajnih brojeva

Generator pseudo-slučajnih brojeva (PRNG), poznat i kao deterministički generator slučajnih brojeva, je algoritam za generiranje niza brojeva čija svojstva odgovaraju svojstvima nizu slučajnih brojeva. Niz pseudo-slučajnih brojeva nije uistinu slučajan zbog činjenice da je u potpunosti određen relativno malim skupom početnih vrijednosti koje se nazivaju sjeme (*eng. seed*) generatora pseudo-slučajnih brojeva, ali sjeme može uključivati potpuno slučajne vrijednosti. Generatori pseduo-slučajnih brojeva su važni u praksi zbog svoje brzine i obnovljivosti, [17].

Generator pseudo-slučajnih brojeva središnji je algoritam za simulacije (npr. Monte Carlo metoda), elektroničke igre (npr. proceduralno generiranje) te kriptografiju. Kriptografski programi zahtijevaju da izlaznu vrijednost nije moguće predvidjeti na temelju prethodnih izlaznih vrijednosti. Uz to zahtijevaju i bolje razrađene algoritme koji ne nasljeđuju linearnost jednostavnih generatora pseudo-slučajnih brojeva. Početno stanje generatora pseudo-slučajnih brojeva može biti proizvoljno odabrano korištenjem sjemena. Svaki puta kada se generator inicijalizira istim sjemenom generirat će se isti niz brojeva, [17].

Glavno otvoreno pitanje teoretske i praktične kriptografije glasi: Postoji li način za razlikovanje izlaznog niza kvalitetnog generatora pseudo-slučajnih brojeva sa pravim slučajnim nizom brojeva bez znanja koji algoritam je korišten i koje je bilo njegovo početno stanje. Sigurnost većine kriptografskih algoritama i protokola temelji se na pretpostavci da nije moguće na jednostavan način, odnosno da je neisplativo, otkriti tu razliku, [17].

6.3.2. Skrivanje poruke nasumičnim LSB algoritmom

Postojeći LSB algoritam nadograđuje se na način da pomoću generatora pseudo-slučajnih brojeva generiraju uređeni parovi nasumičnih brojeva koji određuju koordinate piksela na slici u čije će se RGB komponente spremati bitovi tajne poruke. Pseudo kod algoritma za generiranje pseudo-slučajnih koordinata piksela prikazan je Slika 19.

```
def generiraj_nasumicne_koordinate_piksela(sjeme, slika):
    prng = inicijalizacija_generatora(hash_funkcija(sjeme))
    nasumicni_index_reda = prng.generiraj_bez_ponavljanja(0,broj_redova(slika))
    nasumicni_index_kolone = prng.generiraj_bez_ponavljanja(0,broj_kolona(slika))
    nasumicni_pikseli = prng.generiraj_bez_ponavljanja([nasumicni_index_reda,
                        nasumicni_index_kolone])
    nasumicni pikseli = prng.promjesaj(nasumicni_pikseli)
    vrati nasumicni_pikseli
```

Slika 19. *Pseudo kod algoritma za generiranje slučajnih koordinata piksela*

Algoritam radi na način da se prvo inicijalizira generator pseudo-slučajnih brojeva sa početnim stanjem. Početno stanje predstavlja vrijednost hash funkcije sjemena. Sjeme je tajni ključ koji unosi korisnik, a hash funkcija je funkcija koja pomoću određenog algoritma vraća znakovni niz fiksne dužine koji se razlikuje za različite ulazne vrijednosti. Za potrebe ovog rada korišten je SHA256 hash algoritam. Nakon toga se generira slučajan niz brojeva iz raspona [nula, ukupan_broj_redaka> i [nula, ukupan_broj_kolona> te se nakon toga formira lista uređenih parova koji određuju poziciju piksela na slici. Zadnji korak je još jedno nasumično miješanje liste uređenih parova za dobivanje potpuno slučajnog razmještaja piksela na slici.

```
def nasumicno_lsb_skrivanje(slika, poruka, kljuc):
    index = 0
    sifrat = vigenerovo_sifriranje(poruka, kljuc)
    binarni_zapis = pretvori_u_binarni_zapis(sifrat)
    pikseli = generiraj_nasumicne_koordinate_piksela(kljuc, slika)
    za p u pikseli:
        rgb = dohvati_rgb(p)
        ako je index < duljina(binarni_zapis):
            zamjeni_lsb(rgb,binarni_zapis[index])
            index += 1
    spremi_stego_sliku()
```

Slika 20. *Pseudo kod nasumičnog LSB algoritma za skrivanje poruke*

Nakon ovog koraka proces skrivanja sličan je procesu poboljšanog LSB algoritma. Tajna poruka, odnosno otvoreni tekst šifrira se Vigenèrovom šifrom, a šifrat se u slikovnu datoteku sprema nasumičnim redoslijedom koji je odredio generator pseudo-slučajnih brojeva. Pseudo kod algoritma prikazan je Slika 20.

6.3.3. Izvlačenje poruke nasumičnim LSB algoritmom

Proces izvlačenja poruke nasumičnim LSB algoritmom zahtijeva da se prvo inicijalizira generator pseudo-slučajnih brojeva sa istim sjemenom koje je korišteno prilikom skrivanja slike. Nakon toga se dohvaćaju pikseli onim redoslijedom koji generira generator pseudo-slučajnih brojeva te se čitaju bitovi najmanje važnosti. Dobiveni niz bitova pretvara se u ASCII znakove te se dobiveni šifrat dešifrira Vigenèrovom šifrom. Nakon toga čita se niz znakova do jedinstvene oznake te se ispisuje sadržaj tajne poruke. Pseudo kod algoritma za izvlačenje prikazan je na Slika 21.

```
def nasumicno_lsb_izvlacenje(slika, kljuc):
    pikseli = generiraj_nasumicne_koordinate_piksela(kljuc, slika)
    za p u pikseli:
        rgb = dohvati_rgb(p)
        niz_bitova += procitaj_lsb(rgb)
    sifrat = pretvori_u_ASCII(niz_bitova)
    poruka = vigenerovo_desifriranje(sifrat)
    vrati poruka
```

Slika 21. *Pseudo kod nasumičnog LSB algoritma za izvlačenje poruke*

U prethodnim poglavljima objašnjeno je svojstvo generatora pseudo-slučajnih brojeva koje omogućuje ponovno generiranje istog niza slučajnih brojeva ako je poznata vrijednost sjemena, odnosno njegova inicijalna vrijednost. Bez vrijednosti sjemena napadaču je gotovo nemoguće otkriti kojim redoslijedom je potrebno čitati bitove najmanjih vrijednosti da bi

dobio sadržaj skrivene poruke. Broj mogućih kombinacija spremanja poruke u nasumične piksele slike lako se izračuna korištenjem formule (17) za računanje broja varijacija bez ponavljanja:

$$V_p^n = \frac{n!}{(n-p)!} \tag{17}$$

Gdje je:

- V – varijacija n-tog razreda
- n – uređena n-torka različitih elemenata
- p – skup p članova

Ako za primjer uzmemo sliku rezolucije 640x480 piksela ukupni broj piksela, odnosno n, predstavlja umnožak tih dvaju brojeva i iznosi 307200. Ako poruka koju skrivamo ima 10 znakova, a znamo da svaki znak prikazujemo kombinacijom 8 bitova, potrebno je sakriti 80 bitova. U svaki piksel možemo sakriti 3 bita podataka pa iz toga dobijemo da nam je potrebno 80/3 piksela, odnosno 27 piksela i to je vrijednost p. Uvrštavanjem brojeva u formulu dobiva se:

$$V_p^n = \frac{307200!}{(307200-27)!} = 1{,}44x10^{148}$$

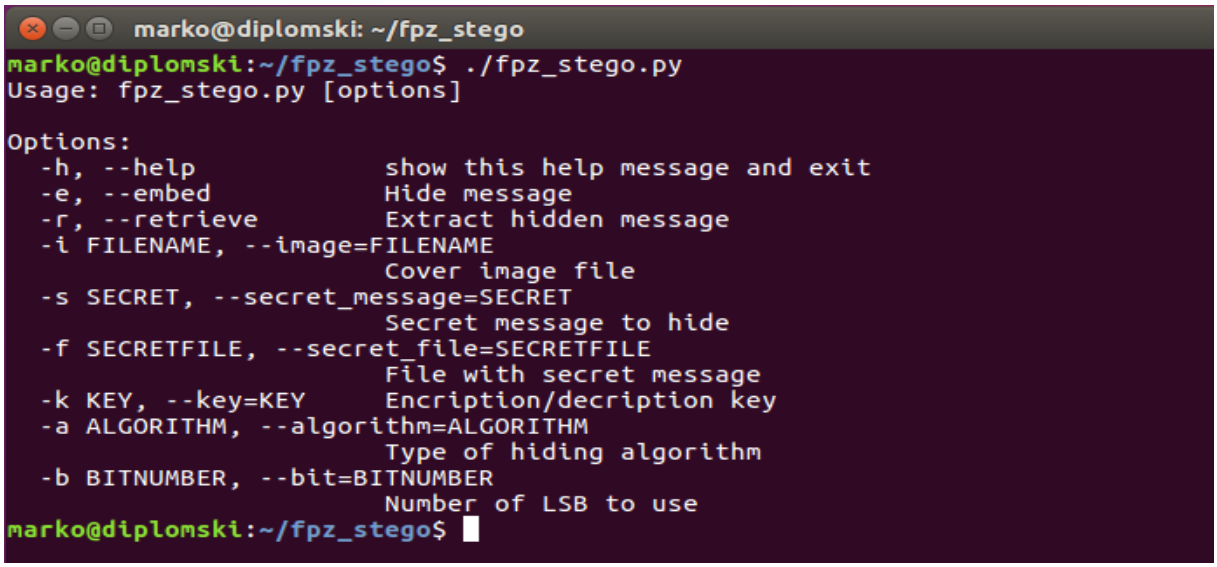
Iz ovog računa lako je zaključiti da se povećanjem duljine poruke broj u nazivniku smanjuje, a samim time i broj varijacija postaje sve veći, odnosno dovoljno je velik da možemo reći kako je nemoguće korištenjem današnjih računala isprobati sve varijacije u razumnom vremenu.

7. KORIŠTENJE ALATA I ANALIZA ALGORITAMA

U prethodnom poglavlju opisana je izrada algoritama pomoću kojih razvijeni stego alat skriva poruku u objekt nositelj. Implementacija algoritama izvedena je u programskom jeziku Python te se u ovom poglavlju daje detaljan pregled korištenja razvijenog stego alata te uspješnost skrivanja i izvlačenja skrivene poruke iz objekta nositelja. Za potrebe testiranja alata korištene se slike *San_Joaquin_Church.png* i *Landscape_Dogon_Mali.png* preuzete se stranice *Wikimedia Commons* (https://commons.wikimedia.org). Slike su objavljene pod licencom otvorenog sadržaja, što znači da se mogu koristiti i mijenjati bez posebne dozvole vlasnika licence.

7.1. Korištenje alata

Alat se pokreće iz komandne linije pozivanjem datoteke fpz_stego.py. Program se izvršava naredbom *„python fpz_stego.py [parametri]“* ili *„./fpz_stego.py [parametri]“*. Pokretanjem programa bez dodatnih parametara ispisuje se uputa kako koristiti program te objašnjenje pojedinog parametra kao što je prikazano na **Slika 22**.



Slika 22. *Pokretanje razvijenog stego alata*

Parametri koji se prosljeđuju programu imaju skraćeni i duži naziv te kratko objašnjenje. Tablica 3 prikazuje naziv i opis parametara koji se mogu proslijediti programu.

Tablica 3. *Popis ulaznih parametara alata s opisom*

Naziv parametra	Opis parametra
-h --help	Ispisuje se uputa o korištenju programa
-e --embed	Označava da se radi o skrivanju poruke u objekt nositelj
-r --retrieve	Označava da se radi o izvlačenju poruke iz stego objekta
-i --image	Naziv slike u koju se skriva poruka ili iz koje se poruka izvlači
-s --secret	Sadržaj tajne poruke koji se označava navodnicima
-f --secretfile	Naziv datoteke u kojoj se nalazi tajna poruka za skrivanje
-k --key	Vrijednost ključa za šifriranje poruke i postavljanja PRNG-a
-a --algorithm	Odabir vrste algoritma kojim će se poruka sakriti u sliku
-b --bit	Broj bitova najmanje važnosti koji se koriste za skrivanje slike

7.1.1. Korištenje alata za skrivanje poruke

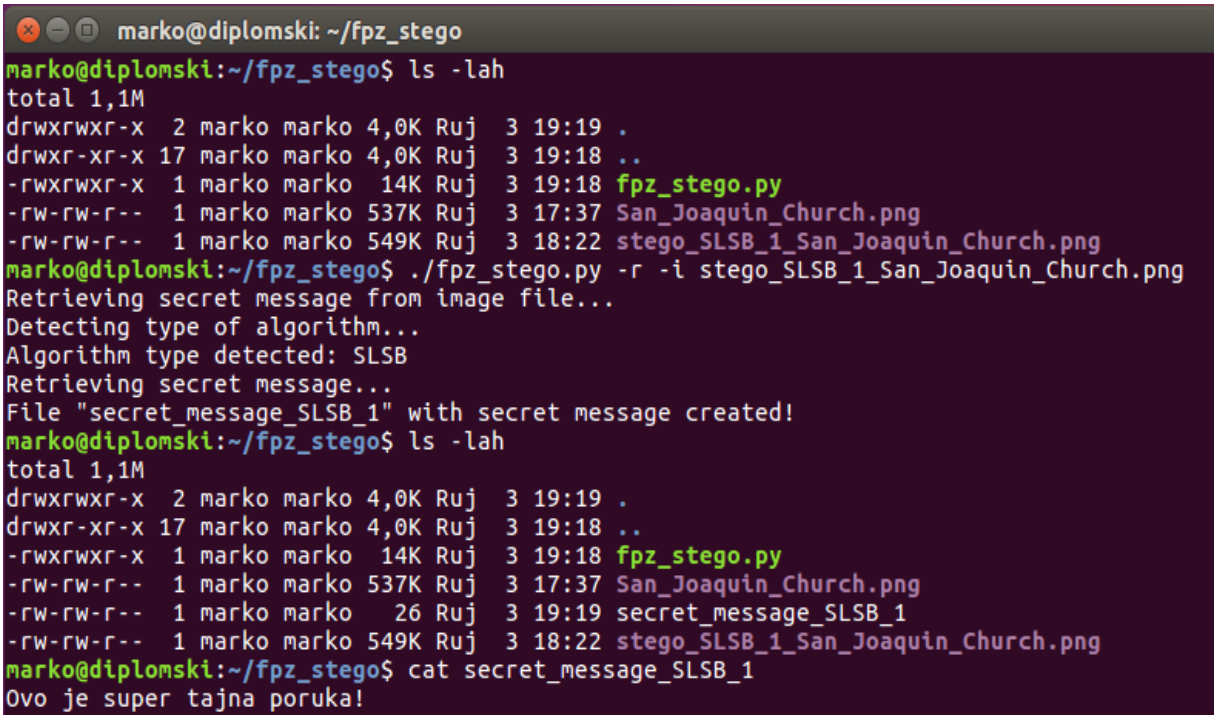
Ako se program koristi za skrivanje poruke u sliku tada su obavezni parametri osim parametra *-e* koji označava da će se program koristiti za skrivanje poruke i parametar *-i* pomoću kojega se navodi naziv slike koja će poslužiti kao objekt nositelj, parametar *-s* ili *-f* koji omogućuje unos tajne poruke te parametar *-a* za odabir algoritma skrivanja. Tajnu poruku moguće je unijeti na dva načina: unosom poruke između znakova navodnika upisivanjem u komandnu liniju nakon parametra *-s* ili navođenjem naziva datoteke u kojoj je zapisana tajna poruka nakon parametra *-f*. Primjer korištenja alata za skrivanje poruke prikazan je Slika 23.

Slika 23. *Primjer korištenja alata za skrivanje poruke*

Dodatni parametar koji se može koristiti prilikom skrivanja poruke u sliku je parametar -b koji označava koliko bitova najmanje važnosti će se koristiti prilikom skrivanja poruke. Taj parametar koristan je za skrivanje poruka dužeg sadržaja, ali treba uzeti u obzir veću vizualnu degradaciju slike.

7.1.2. Korištenje alata za izvlačenje poruke

Ako se program koristi za izvlačenje tajne poruke iz slike tada su obavezni parametri osim parametra *-r* koji označava da će se program koristiti za izvlačenje poruke iz slike i parametar *-i* pomoću kojeg se navodi naziv stego slike iz koje se izvlači tajna poruka. Budući da se prilikom skrivanja poruke na početak sadržaja tajne poruke postavi oznaka algoritma kojim je poruka skrivena, program će sam detektirati na koji način će izvući poruku pa nije potrebno eksplicitno navoditi parametar *-a*. Međutim, ako je prilikom skrivanja poruke odabran veći broj bitova najmanje važnosti u koje će se skrivati tajna poruka tada je potrebno obavezno navesti broj bitova koristeći parametar *-b*. Nakon što program završi sa procesom izvlačenja tajne poruke kreirati će tekstualnu datoteku sa sadržajem tajne poruke. Primjer korištenja programa za izvlačenja tajne poruke prikazan je Slika 24.



Slika 24. *Primjer korištenja alata za izvlačenje poruke*

Originalna slika i stego objekt sa skrivenom porukom vizualno se ne razlikuju, odnosno ljudsko oko nije sposobno prepoznati razlike. Originalna slika i stego objekt prikazani su na Slika 25 i Slika 26.

Slika 25. *Originalna slika u korištenom primjeru*
Izvor:
https://commons.wikimedia.org/wiki/File:San_Joaquin_Church_2.png

Slika 26. *Stego objekt sa skrivenom porukom*

7.2. Analiza razvijenih algoritama

Za potrebe testiranja alata generirano je sedam tekstualnih datoteka različitih veličina. Svaka tekstualna datoteka odgovara maksimalnoj duljini poruke koja se skriva u sliku, a ovisi o broju bitova najmanje važnosti koji se koriste. Iz tog razloga napisana je pomoćna skripta koja za odabranu sliku računa maksimalni broj bitova koje je moguće pohraniti te nakon toga stvara tekstualnu datoteku generičkog sadržaja *lorem ipsum*.

U odabranu sliku skrivana je poruka različite duljine korištenjem različitog broja bitova najmanje važnosti. Detaljan pregled veličine slike i količine podataka skrivenih u nju prikazan je Tablica 4.

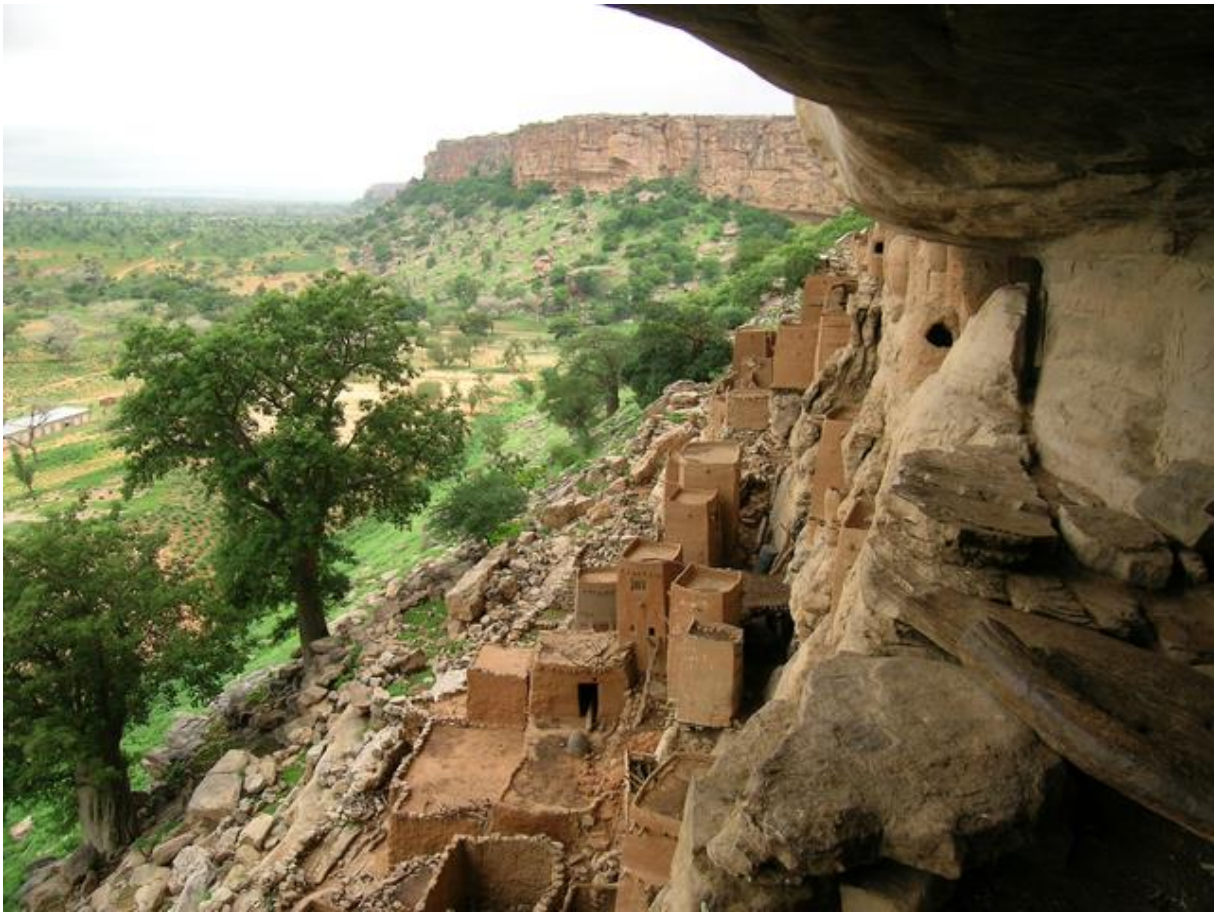
Tablica 4. *Duljina skrivene poruke ovisno o broju korištenih bitova*

Dimenzija slike	Broj bitova najmanje važnosti koji se koriste	Maksimalna duljina spremljene poruke [bytes]	Duljina spremljene poruke [bytes]
640x480	1	115200	115128
640x480	2	230400	230268
640x480	3	345600	345026
640x480	4	460800	460520
640x480	5	576000	575431
640x480	6	691200	690892
640x480	7	806400	806157

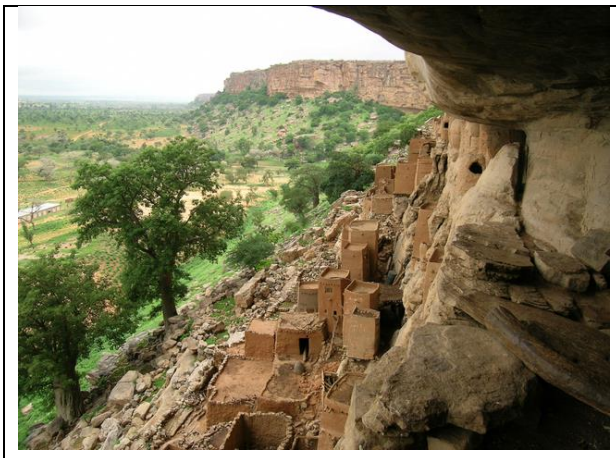
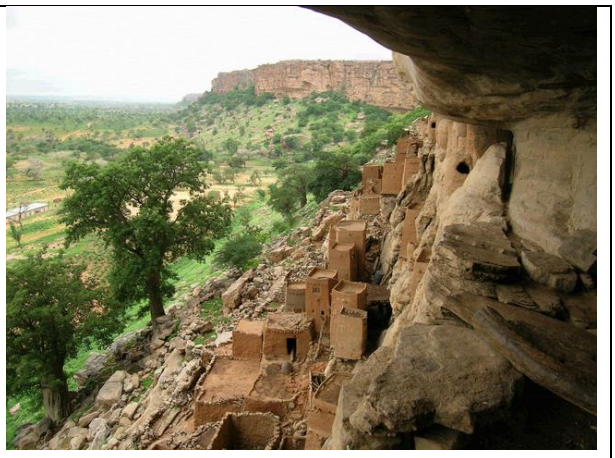
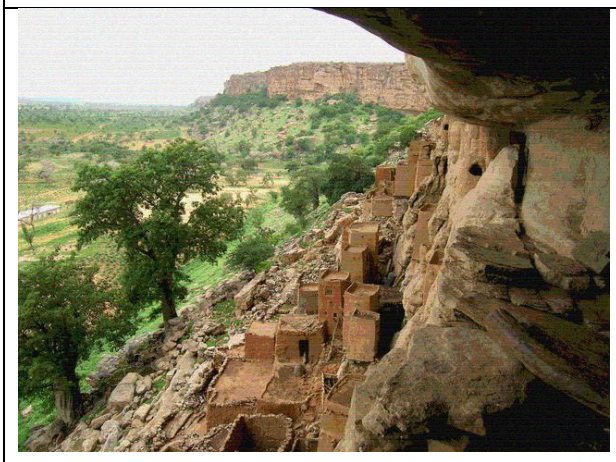
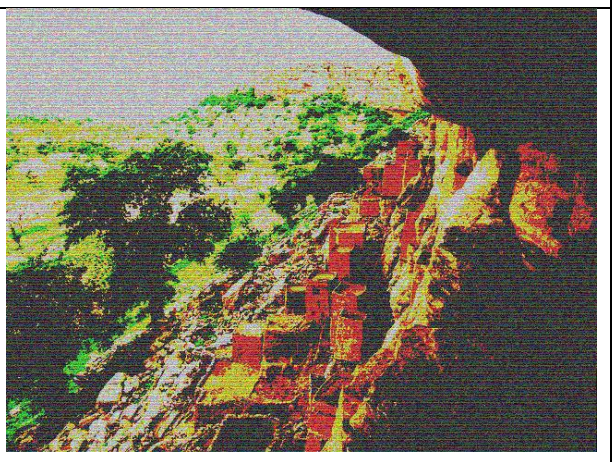
7.2.1. Vizualna detekcija

Slika 27 prikazuje originalnu sliku prije skrivanja poruke u nju, dok je na Slika 28, Slika **29** i Slika **30** prikazan utjecaj korištenja različitog broja bitova najmanje važnosti primjenom sva tri razvijena algoritma te njihova robusnost na vizualne napade, odnosno koliko bitova se može koristiti bez da ljudsko oko primijeti veću degradaciju slike.

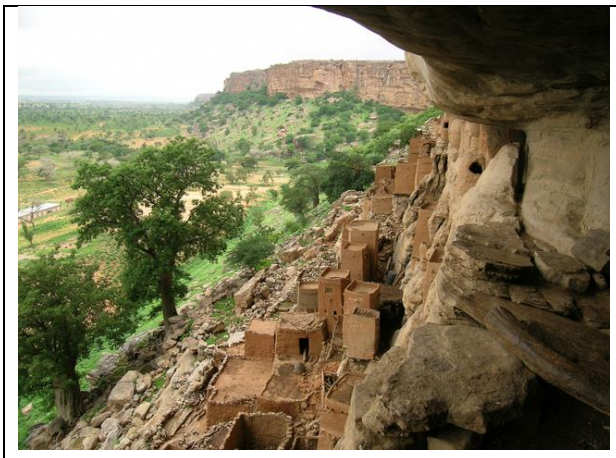
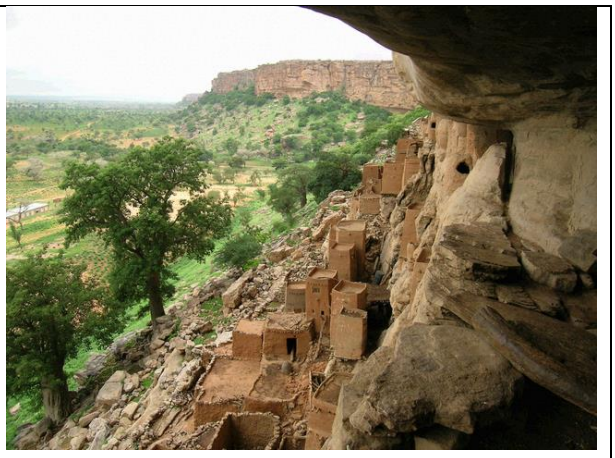
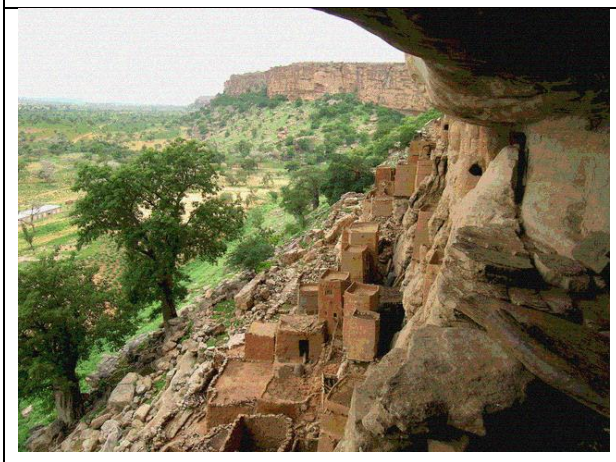
Na slikama je vidljivo da korištenjem do 3 bita najmanje važnosti ljudsko oko ne može primijetiti degradaciju. Korištenjem većeg broja bitova najmanje važnosti vizualna degradacija je uočljiva ljudskim okom. Također je uočljiv različiti uzorak degradacije slike kod ELSB i RLSB algoritma. Kod primjene SLSB i ELSB algoritma nastaju vodoravne linije, dok kod RLSB-a nastaje jednoliki točkasti šum.



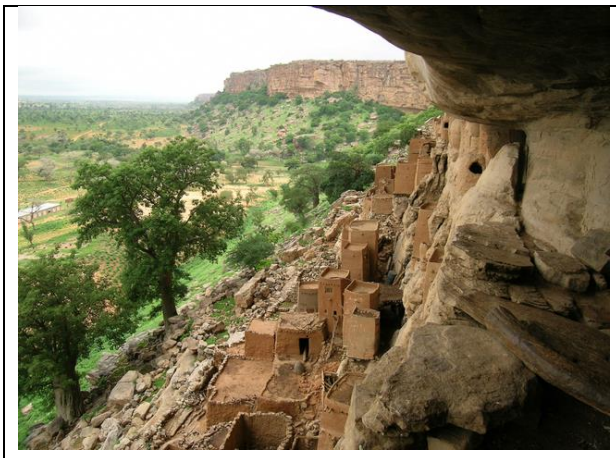
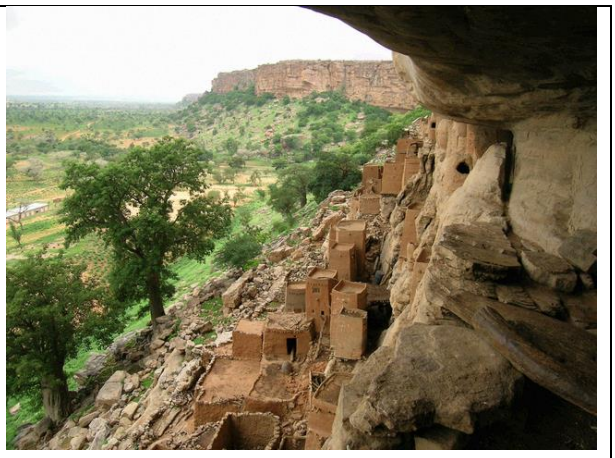
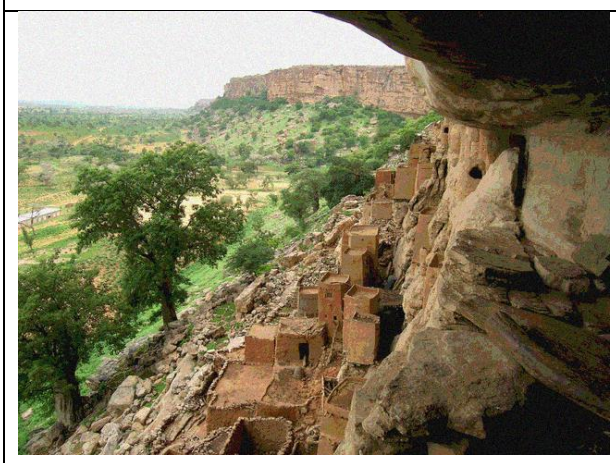
Slika 27. *Originalna slika prije skrivanja poruke*
Izvor: https://commons.wikimedia.org/wiki/File:Landscape_Dogon_Mali.png

	
koristi se 1 bit najmanje važnosti za skrivanje poruke	koriste se 3 bita najmanje važnosti za skrivanje poruke
	
koristi se 5 bitova najmanje važnosti za skrivanje poruke	koristi se 7 bitova najmanje važnosti za skrivanje poruke

Slika 28. *Otpornost jednostavnog LSB algoritma na vizualne napade*

	
koristi se 1 bit najmanje važnosti za skrivanje poruke	koriste se 3 bita najmanje važnosti za skrivanje poruke
	
koristi se 5 bitova najmanje važnosti za skrivanje poruke	koristi se 7 bitova najmanje važnosti za skrivanje poruke

Slika 29. *Otpornost poboljšanog LSB algoritma na vizualne napade*

	
koristi se 1 bit najmanje važnosti za skrivanje poruke	koriste se 3 bita najmanje važnosti za skrivanje poruke
	
koristi se 5 bitova najmanje važnosti za skrivanje poruke	koristi se 7 bitova najmanje važnosti za skrivanje poruke

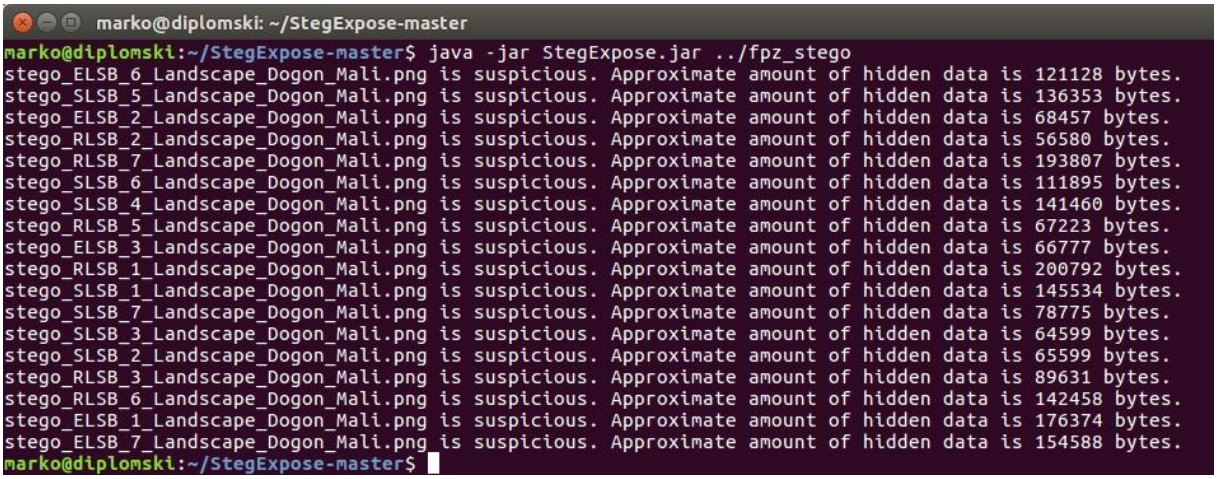
Slika 30. *Otpornost nasumičnog LSB algoritma na vizualne napade*

7.2.2. Otkrivanje neobičnih uzoraka

Za otkrivanje neobičnih uzoraka korišten je jednostavan alat za steganalizu *StegExpose*. Alat je otvorenog koda i razvijen je posebno za detekciju stego objekata u koje je poruka skrivena primjenom LSB algoritama u slikama bez gubitaka.

Na Slika 31 prikazano je pokretanje StegExpose alata te dobiveni rezultati. Od dvadesetijednog stego objekta, StegExpose je detektirao osamnaest potencijalno sumnjivih te je uz svaku datoteku naveo procjenu količine skrivenih podataka.

Rezultat testa bio je očekivan jer je poznato da su ovakve metode skrivanja poruke ranjive na detekciju statističkim testovima te je lako utvrditi postojanje skrivene poruke. Procjena veličine skrivene poruke uvelike odstupa od stvarne vrijednosti.



Slika 31. *Rezultati detekcije stego objekata korištenjem alata StegExpose*

8. ZAKLJUČAK

Razvijeni alat u ovom diplomskom radu omogućuje skrivanje i izvlačenje informacija iz slikovne datoteke bez gubitaka korištenjem LSB algoritma. Na temelju provedenih testova može se zaključiti da su stego objekti kreirani ovim alatom otporni na vizualne napade ako se koristi manje od četiri bita najmanje važnosti, jer je vizualna degradacija slike neprimjetna ljudskom oku.

Skrivanje informacije korištenjem LSB algoritma lako se detektira raznim statističkim metodama, što je i pokazano analizom kreiranih stego objekata steganalitičkim alatom StegExpose koji je detektirao potencijalne stego objekte uz visoki postotak detekcije od gotovo 86%.

Skrivanjem informacija u slikovnu datoteku primjenom razvijenog nasumičnog LSB algoritma ostvaruje se visoki stupanj sigurnosti informacije. Ako napadač i posumnja u postojanje skrivene informacije gotovo da mu je nemoguće istu izvući u razumnom vremenu, zbog velikog broja mogućih kombinacija zapisivanja poruke u sliku.

Iz prikazanih analiza možemo zaključiti da steganografske metode omogućuju pouzdanu zaštitu tajnosti informacija te predstavljaju odličnu alternativu kriptografiji. Kombinacijom obje discipline postiže se visoki stupanj robusnosti, jer ako napadač i uspije izvući informacije iz objekta one su i dalje nerazumljive i nečitljive. Objekti u koje se informacija skriva uobičajeni su u svakodnevnoj komunikaciji pa je napadaču teško analizirati svaki objekt koji prođe komunikacijskim kanalom.

Iako je ostvaren visoki stupanj tajnosti informacije ovaj alat koristi osnovne kriptografske algoritme pa bi se u daljnjem razvoju svakako trebalo implementirati provjereni jaki kriptografski algoritmi, npr. AES ili DES. Također bi bilo dobro dodati funkcionalnost skrivanja informacije u slikovne datoteke s gubicima, te implementirati algoritme za skrivanje informacija u druge objekte nositelje.

LITERATURA

[1] Schneier, B.: *Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition*, John Wiley & Sons Inc., United States of America, 1996.

[2] Žubrinić, D.: *Diskretna matematika*, Element, Zagreb, 2002.

[3] URL: http://en.wikipedia.org/wiki/Cryptography (pristupljeno: 6. rujna 2016.)

[4] URL: http://web.math.pmf.unizg.hr/~duje/kript/kriptografija.html (pristupljeno: 6. rujna 2016.)

[5] Hrvatska akademska i istraživačka mreža, CARNet: Steganografija, Zagreb, 2006. URL: http://www.cert.hr/sites/default/files/CCERT-PUBDOC-2006-04-154.pdf (pristupljeno: 6. rujna 2016.)

[6] URL: https://en.wikipedia.org/wiki/Steganography (pristupljeno: 6. rujna 2016.)

[7] Kipper, G.: *Investigator's guide to steganography*, Auerbach Publications, London 2004.

[8] Arsenović, M., Avanić A., Kunšt, D., Kranjčec, T.: *Steganografija dokumentacija*, Fakultet elektrotehnike i računarstva, Zagreb, 2011. URL: https://www.fer.unizg.hr/_download/repository/Steganografija_-_dokumentacija.pdf (pristupljeno: 6. rujna 2016.)

[9] Katzenbeisser, S., Petitcolas, F. A. P.: *Information hiding techniques for steganography and digital watermarking*, Artech House Inc, Norwood, 2000.

[10] EC-Council: *Computer Forensics, Investigating Data and Image Files*, Cengage Learning, Clifton Park, NY USA

[11] URL: http://en.wikipedia.org/wiki/Steganalysis (pristupljeno: 6. rujna 2016.)

[12] Kaur, A., Kaur, R., Kumar, N.: *A Review on Image Steganography Techniques*, International Journal of Computer Applications, Volume 123 – Number 4, 38-42, 2015.

[13] Morkel, T., Eloff, J.H.P., Olivier, M.S.: *An Overview of Image Steganography*, Proceedings of the fifth annual Information Security South Africa Conference, Sandton, 2005.

[14] URL: http://e.math.hr/stegano/stegano_print.html (pristupljeno: 6. rujna 2016.)

[15] URL: http://www.petitcolas.net/steganography/image_downgrading/ (pristupljeno: 6. rujna 2016.)

[16] URL: http://www.vcl.fer.hr/dtv/jpeg/wave.htm (pristupljeno: 6. rujna 2016.)

[17] URL: https://en.wikipedia.org/wiki/Pseudorandom_number_generator (pristupljeno: 6. rujna 2016.)
[18] URL: http://e.math.hr/old/vigenere/index.html (pristupljeno: 6. rujna 2016.)
[19] URL: https://en.wikipedia.org/wiki/Vigenère_cipher (pristupljeno: 6. rujna 2016.)

POPIS KRATICA

ASCII	(American Standard Code for Information Interchange) Američki standardni znakovnik za razmjenu informacija
AES	(Advanced Encryption Standard) američki državni standard za šifriranje
BMP	(Bitmap Image File) nekompresirani slikovni format datoteke
DCT	(Discrete Cosine Transform) diskretna kosinusna transformacija
DES	(Data Encryption Standard) standardni federalni kriptografski algoritam u SAD-u
DFT	(Discrete Fourier transform) diskretna Fourierova transformacija
DSSS	(Direct Sequence Spread Spectrum) prošireni spektar direktnim slijedom
DWT	(Discrete Wavelet Transform) diskretna valna transformacija
ELSB	(Enhanced LSB Algorithm) poboljšani LSB algoritam
FHSS	(Frequency Hopping Spread Spectrum) prošireni spektar s frekvencijskim skokom
GIF	(Graphical Interchange Format) 8-bitni grafički format slikovnog zapisa
JPEG	(Joint Photographic Experts Group) komprimirani slikovni format s gubicima
LSB	(Least Significant Bit) bit najmanje važnosti
LTS	(Long Term Support) dugoročna podrška
PIL	(Python Image Library) Pythonova biblioteka za obradu slikovnih datoteka
PRNG	(Pseudo-Random Number Generator) generator pseudo-slučajnih brojeva
RGB	(Red Green Blue) prostor boja definiran pomoću tri osnovne boje: crvena, zelena i plava
RLSB	(Random LSB Algorithm) nasumični LSB algoritam
SHA	(Secure Hash Algorithm) siguran algoritam sažimanja
SLSB	(Simple LSB Algorithm) jednostavni LSB algoritam
YUV	(Luminance (Y) Blue–Luminance (U) Red–Luminance (V)) prostor boja definiran pomoću jedne komponente za svjetlinu te dvije komponente za svjetlinu plave i crvene boje

POPIS SLIKA

POPIS TABLICA

METAPODACI

Naslov rada: Skrivanje informacija u objekt nositelj korištenjem LSB algoritma

Student: Marko Stanec

Mentor: prof. dr. sc. Dragan Peraković

Naslov na drugom jeziku (engleski):

Hiding data in carrier object using LSB algorithm

Povjerenstvo za obranu:

- prof. dr. sc. Zvonko Kavran predsjednik
- prof. dr. sc. Dragan Peraković mentor
- doc. dr. sc. Ivan Grgurević član
- doc. dr. sc. Marko Periša zamjena

Ustanova koja je dodijelila akademski stupanj: Fakultet prometnih znanosti Sveučilišta u Zagrebu

Zavod: Informacijsko-komunikacijski promet

Vrsta studija: diplomski

Studij: Promet

Datum obrane diplomskog rada: 27. rujna 2016.

Sveučilište u Zagrebu
Fakultet prometnih
znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj diplomski rad isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu diplomskog rada pod naslovom **Skrivanje informacija u objekt nositelj korištenjem LSB algoritma**

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

U Zagrebu, 6.9.2016

Student:

(potpis)