UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET

Steganografija zasnovana na LSB algoritmu

Seminarski rad

Digitalna forenzika

Studijski program:

Računarstvo i informatika

Modul:

Softversko inženjerstvo

Student:

Katarina Lazarević, br. ind 1995

Predmetni nastavnik:

Prof. Dr. Bratislav Predić

SADRŽAJ

1. UVOD

U današnjem digitalnom dobu, gde su informacije postale najvredniji resurs, zaštita podataka ima izuzetno veliki značaj. Pored uobičajenih metoda šifrovanja (kriptografije), sve veću pažnju privlače i tehnike skrivanja samog postojanja poruke – **steganografija**.

Za razliku od kriptografije, čiji je cilj da zaštiti sadržaj poruke tako da ona ne bude razumljiva neovlašćenim licima, **steganografija** ima zadatak da sakrije činjenicu da poruka uopšte postoji. Na taj način komunikacija ostaje neprimetna, a poverljivi podaci se mogu prenositi kroz naizgled obične slike, zvukove ili video zapise.

Jedna od najpoznatijih i najčešće korišćenih metoda digitalne steganografije je **Least Significant Bit (LSB)** tehnika. Ova metoda koristi činjenicu da promena najmanje značajnog bita u pikselu slike ne utiče vidno na njen izgled, ali omogućava sakrivanje velikog broja podataka.

LSB pristup je jednostavan, efikasan i pogodan za implementaciju u različitim programskim jezicima, posebno u Pythonu koji nudi bogatu podršku za obradu slika kroz biblioteke poput **Pillow** i **NumPy**.

Cilj ovog rada je da se prikaže princip rada LSB steganografije i da se kroz praktičnu implementaciju demonstrira kako se tekstualna poruka može sakriti unutar digitalne slike i kasnije uspešno rekonstruisati, bez vidljivih promena u izgledu slike.

U okviru implementacije biće razvijen Python program koji omogućava:

- unos poruke koju korisnik želi da sakrije,
- kodiranje poruke u okviru slike,
- dekodiranje poruke iz kodirane slike,
- kao i vizuelno poređenje originalne i modifikovane slike radi prikaza neprimetnosti promena.

2. STEGANOGRAFIJA

Steganografija predstavlja oblast informacione bezbednosti koja se bavi prikrivanjem postojanja informacija. Za razliku od kriptografije, koja samo maskira značenje poruke, steganografija ide korak dalje – njen cilj je da komunikacija uopšte ne bude prepoznata kao takva.

Ideja steganografije je stara koliko i sama potreba ljudi da tajno komuniciraju. Još u antičkoj Grčkoj poruke su se skrivale ispod slojeva voska na drvenim pločicama ili urezivale na kožu, dok su se u modernom dobu razvile digitalne tehnike skrivanja informacija unutar različitih medija.

U digitalnom kontekstu, steganografija koristi fajlove kao "nosioca" tajnih poruka (eng. *cover object*), a rezultat skrivanja naziva se *stego-object*.Poruka koja se krije može biti tekst, binarni fajl, slika, ili bilo koji niz podataka, dok medijum za prenos može biti slika, zvuk, video zapis ili mrežni paket.

Osnovni cilj je da modifikacije koje nastaju tokom skrivanja budu **neprimetne za ljudsko oko i sluh**, ali da se sakriveni podaci kasnije mogu **pouzdano rekonstruisati**.

Ključni pojmovi u steganografiji:

- Cover Medium (Pokrivač): To je nosilac informacija u koji se skrivaju podaci. To može biti slika, zvuk, tekst, video ili bilo koji drugi format podataka.
- Hidden Data (Skriveni Podaci): To su podaci koje želite sakriti unutar pokrivača. Ovi podaci mogu biti tekst, slike, ili bilo koji drugi format.
- Stego Key (Stego Ključ): Ovaj ključ je potreban da bi se skriveni podaci izvadili iz pokrivača. On funkcioniše slično kao ključ u kriptografiji.
- Steganografski Algoritmi: To su postupci ili metode koji se koriste za skrivanje podataka. Postoji mnogo različitih pristupa steganografiji, uključujući manipulaciju boja piksela u slikama, izmene audio signala ili čak ubacivanje tajnih poruka u tekst tako da ne privlače pažnju.

2.1. Princip rada steganografije

Proces steganografije obično se sastoji iz tri faze:

1. **Kodiranje (Embedding)** – u ovoj fazi tajna poruka se ubacuje u odabrani medijum pomoću određenog algoritma.

2. **Prenos (Transmission)** – medijum sa sakrivenom porukom šalje se primaocu putem interneta, e-maila ili drugog kanala komunikacije.

3. **Dekodiranje (Extraction)** – primalac koristi odgovarajući algoritam da iz medijuma izdvoji skrivene podatke.

Za uspešnu implementaciju steganografije bitno je da se postigne balans između tri ključna faktora:

- **Neprimetnost (imperceptibility)** – modifikacije ne smeju biti uočljive ljudskom oku ili uhu.

- **Kapacitet (capacity)** – količina podataka koja se može sakriti u okviru jednog medijuma.

- **Otpornost (robustness)** – sposobnost sistema da zadrži skrivene podatke čak i nakon kompresije, skaliranja ili drugih promena.

U steganografiji postoje dve vrste poruka: prva je „kontejnerska“ poruka, a druga je tajna poruka, pri čemu jedna ima zadatak da sakrije sadržaj druge, kako bi je učinila nevidljivom za bilo koji prisluškivač. Generalno, skrivene poruke izgledaju (ili su deo) nečeg drugog: slike, članci, liste ili drugi naslovni tekst. Na primer, skrivena poruka može biti nevidljivo mastilo između redova privatnog pisma.
Postoje dva glavna steganografska modela: **injektivna** steganografija i **generativna** steganografija. Najviše se koristi injektivna steganografija, sastoji se od ubacivanja (ubrizgavanja) tajne poruke u drugu poruku koja deluje kao kontejner, kako ne bi bila vidljiva ljudskom oku i da se praktično ne bi razlikovala od originala. Kod generativne steganografije, umesto tradicionalnog pristupa gde se uzima postojeći kontejner i u njega ubacuje poruka, kreira se novi kontejner, kako bi se poruka sakrila na najbolji mogući način.

Steganografija kao tehniku najčešće koristi substituciju. Većina komunikacionih kanala (telefonske linije, radio-prenosi, itd.) emituju signale koji su uvek praćeni nekom vrstom šuma. Ovaj šum se može zameniti signalom – tajnom porukom – koja je transformisana na takav način da se, osim ako ne znate tajni ključ, ne razlikuje od stvarne buke, pa se poruka može prenositi bez izazivanja sumnje.

Steganografija nije zamena za kriptografiju. Dok steganografija skriva postojanje podataka, kriptografija ih štiti od neovlašćenog pristupa. Kombinacija ove dve tehnike može pružiti dodatni nivo sigurnosti u komunikaciji. Pregled steganografskih algoritama od najjedonstavnijih do kompleksnih:

- Least Significant Bit (LSB) Substitution:
    - Ovo je jednostavan algoritam koji koristi najmanje značajne bitove (LSB) u pikselima slike za skrivanje informacija. Male promene u boji piksela često su neprimetne ljudskom oku.
- Algoritmi zasnovani na maskiranju i filtriranju
    - Ovi algoritmi su nešto napredniji i koriste se u slikama visoke kvaliteta, često u **formatima bez gubitaka (lossless)**. Umesto direktne izmene bitova, informacija se maskira u delovima slike koji su manje uočljivi ljudskom oku (na primer, u tamnijim ili teksturiranim delovima slike).
    - Ova metoda je otpornija na osnovne analize, ali zahteva preciznije poznavanje karakteristika slike i percepcije ljudskog vida.
- Transformacioni algoritmi (DCT, DWT, DFT)
    - Kod transformacionih algoritama, poruka se ne upisuje direktno u piksele, već u **frekvencijski domen slike**. To znači da se najpre primeni matematička transformacija kao što su:
    - **DCT (Discrete Cosine Transform)** – koristi se u JPEG slikama; poruka se ubacuje u koeficijente srednjih frekvencija.

    - **DWT (Discrete Wavelet Transform)** – deli sliku na više podopsegova i ubacuje informaciju u talasne komponente.

    - **DFT (Discrete Fourier Transform)** – koristi frekvencijske karakteristike slike.

        - Ovi algoritmi su znatno otporniji na kompresiju, rotaciju i filtriranje, jer su promene urađene u prostoru koji nije direktno vidljiv, već u spektralnom predstavljanju slike. Međutim, njihova implementacija je složenija i računski zahtevnija.
- Steganografija zasnovana na veštačkoj inteligenciji
    - Sa razvojem mašinskog učenja i dubokih neuronskih mreža, pojavili su se i **AI-bazirani steganografski sistemi**. Oni koriste **konvolucione neuronske mreže (CNN)** za učenje optimalnih načina skrivanja i ekstrakcije informacija iz slike.
    - Takvi sistemi mogu automatski pronaći delove slike gde su promene najmanje uočljive, kao i rekonstruisati skrivenu poruku čak i nakon značajnih oštećenja slike. Ovaj pristup predstavlja najsavremeniji i najkompleksniji oblik steganografije danas.

3. LSB ALGORITAM

LSB tehnika je najjednostavnija i najraširenija metoda steganografije u prostornom domenu za rastarske slike. Princip se zasniva na činjenici da promena najmanje značajnog bita (LSB) u digitalnom prikazu boje (8-bitni kanal: 0–255) dovodi do veoma male numeričke promene u vrednosti kanala, koja je u većini slučajeva neprimetna ljudskom oku. Korišćenjem poslednjeg bita svakog kanala piksela moguće je sekvencijalno upisati bitove tajne poruke.

Neka je slika dimenzija W×H piksela, i svaki piksel ima tri kanala (R, G, B), svaka vrednost u opsegu [0,255][0,255][0,255]. Ukupan broj bitova dostupnih za ugradnju podataka u čistom LSB pristupu, kada se koristi samo jedan LSB po kanalu, iznosi:

$$kapacitet\_bit = W \times H \times 3$$

Broj bajtova koji se mogu sakriti je:

$$\text{kapacitet\_bajta} = \frac{W \times H \times 3}{8}$$

Za dato bajtno polje poruke M=(m1,m2,…,mn) , binarni tok poruke B se konstruiše kao niz bitova b0b1…b8n−1. Ugradnja radi tako da se za redom zameni LSB svake kanalske vrednosti sa uzastopnim bitom iz B.

Dekodiranje je jednostavno čitanje LSB-ova iz iste sekvence kanala i rekonstruisanje bajtova dok se ne detektuje unapred dogovoreni **end-marker** ili dok se ne pročita unapred poznati broj bajtova.

LSB algoritam ima određena ograničenja. **Kapacitet** zavisi direktno od rezolucije slike; velika slika znači veći prostor za podatke. Ako se koristi jedan LSB po kanalu, **vizuelna degradacija** je praktično neprimetna. Ako se menja više od 1 LSB (npr. 2 LSB), kapacitet raste, ali i vidljivost promena raste eksponencijalno. **Otpornost** - SB je slab na transformacije koje menjaju piksele (kompresija sa gubitkom - JPEG), skaliranje, rotaciju i filtriranje. Zato se LSB najčešće koristi za medije koji neće biti obrađivani nakon ugradnje.

4. IMPLEMENTACIJA

Implementacija LSB steganografskog sistema razvijena je u programskom jeziku Python. Python je odabran kao programski jezik iz nekoliko ključnih razloga:

- Python nudi moćne biblioteke za obradu slika (Pillow) i numeričke operacije (NumPy)
- Sintaksa Python-a omogućava jasnu implementaciju algoritama,
- Brzo prototipovanje i testiranje različitih pristupa

Ključne biblioteke korišćene u implementaciji su:

- **Pillow (PIL Fork) v10.x** - Za učitavanje, manipulaciju i čuvanje slika. Podržava širok spektar formata (PNG, JPEG, BMP, TIFF)
- **NumPy v1.24+** - Za efikasnu manipulaciju piksel podacima kroz vektorske operacije, što drastično ubrzava proces enkodovanja i dekodovanja
- **argparse** - Za kreiranje command-line interfejsa sa profesionalnim rukovanjem argumentima

Sistem je organizovan kroz sledeće komponente:

```
LSB-Steganography/
├── main.py                # Glavna aplikacija
├── LSBSteganography       # Osnovna klasa
│   ├── _text_to_binary()  # Konverzija tekst → binarno
│   ├── _binary_to_text()  # Konverzija binarno → tekst
│   ├── _max_bytes_capacity() # Računanje kapaciteta
│   ├── encode()           # Enkodovanje poruke
│   ├── decode()           # Dekodovanje poruke
│   └── compare_images()   # Analiza razlika
├── create_demo_image()    # Generator test slika
└── main()                 # CLI interfejs
```

Prvi korak u procesu enkodovanja je učitavanje ciljne slike i njena konverzija u odgovarajući format:

```python
img = Image.open(image_path)
if img.mode != 'RGB':
    img = img.convert('RGB')
```

Ova konverzija je kritična iz sledećih razloga:

1. **Transparencija** - PNG slike često sadrže alpha kanal (RGBA format), što bi komplikovalo enkodovanje
2. **Paletne slike** - Neki formati koriste indeksne boje umesto direktnih RGB vrednosti
3. **Grayscale slike** - Crno-bele slike imaju samo jedan kanal, što drastično smanjuje kapacitet

Konverzijom u RGB obezbeđujemo konzistentan radni format sa tačno 3 bajta po pikselu (po jedan za crvenu, zelenu i plavu komponentu).

Pre enkodovanja, sistem mora da proveri da li poruka može stati u odabranu sliku. Kapacitet se računa prema formuli:

Kapacitet (bajtova) = (Širina × Visina × 3 kanala) / 8 bitova - Delimiter_veličina

```
width, height = img.size
total_bits = width * height * 3
delimiter_bits = len(self.delimiter.encode('utf-8')) * 8
available_bits = total_bits - delimiter_bits
max_bytes = available_bits // 8
```

Na primer, za sliku dimenzija 800×600 piksela:

- Ukupno piksela: 800 × 600 = 480,000
- Ukupno bitova: 480,000 × 3 = 1,440,000
- Kapacitet: 1,440,000 / 8 = **180,000 bajtova** (~180 KB teksta)

Što je slika veća, to više podataka može sakriti. Međutim, važno je napomenuti da praktični kapacitet zavisi i od drugih faktora kao što su format slike i njena kompresija.

Za efikasnu manipulaciju pikselima, sliku konvertujemo u NumPy array i "flatten" je u 1D niz:

```
img_array = np.array(img)
height, width, channels = img_array.shape
flat_img = img_array.flatten()
```

Ova transformacija omogućava:

- **Brzinu** - NumPy operacije su 10-100× brže od Python petlji
- **Jednostavnost** - Iteracija kroz piksele postaje linearna
- **Memorijsku efikasnost** - Direktna manipulacija memorijom

4.1. Proces enkodiranja

Enkodiranje kod LSB algoritma predstavlja modifikaciju najmanje značajnog bita svakog bajta. Ovo se postiže kroz dve bitske operacije:

```
for i, bit in enumerate(binary_message):
    flat_img[i] = (flat_img[i] & 0xFE) | int(bit)
```

Analiza operacija

1. `& 0xFE` - AND sa `11111110` (maskiranje)
   - Postavlja LSB na 0, ostali bitovi ostaju nepromenjeni
   - Primer: `11010111 & 11111110 = 11010110`
2. `| int(bit)` - OR sa novim bitom
   - Ako je bit = 1: postavlja LSB na 1
   - Ako je bit = 0: LSB ostaje 0
   - Primer: `11010110 | 00000001 = 11010111`

```
Originalni piksel RGB(214, 198, 142):
R: 11010110 (214) → 11010111 (215) [bit=1]
G: 11000110 (198) → 11000110 (198) [bit=0]
B: 10001110 (142) → 10001111 (143) [bit=1]

Rezultat: RGB(215, 198, 143)
Razlika: ΔR=+1, ΔG=0, ΔB=+1
```

Ove minimalne promene (±1) su vizuelno neuočljive ljudskim okom, što je osnovna prednost LSB tehnike. Nakon enkodovanja, niz se vraća u 3D strukturu i čuva kao slika.

4.2. Proces dekodiranja poruke

Dekodovanje je inverzni proces koji ekstraktuje sakrivenu poruku iz slike. Proces počinje učitavanjem slike i ekstrakcijom LSB iz svakog bajta. Delimiter za kraj poruke je ključan u ovom procesu iz razloga što sprečava čitanje random bitova kao dela poruke time što nedvosmisleno označava kraj. U slučaju da se on ne koristi, sistem bi pokušao da dekoduje **sve** LSB bitove u slici, što bi rezultiralo porukom sa random "garbage" karakterima na kraju.

4.3. Merenje vizuelnih razlika

Nakon što se tajna poruka ugradi u sliku, jedan od osnovnih zadataka steganografije je da se proceni koliko se **kvalitet slike promenio** u odnosu na original.

Cilj je da razlika bude što manja, tj. da **kodirana slika vizuelno bude identična originalnoj**. Sistem implementira kvantitativnu analizu kvaliteta enkodovane slike kroz dve ključne metrike:

1. **MSE (Mean Squared Error) -** MSE predstavlja prosečno kvadratno odstupanje između piksela originalne slike i slike koja sadrži skrivenu poruku.
    a. **Manja vrednost MSE** znači da su razlike između originalne i kodirane slike manje, odnosno da je skrivanje poruke kvalitetnije i manje uočljivo.
    b. Tipične vrednosti MSE za dobre steganografske sisteme kreću se **blizu nule**, što označava minimalnu degradaciju slike.
2. **PSNR (Peak Signal-to-Noise Ratio) -** Metrika koja se često koristi za merenje kvaliteta stereo-slike(slika sa sakrivenim podacima) u poređenju sa originalnom slikom. Koristi se kao dopunska mera uz MSE i izražava se u **decibelima (dB)**.
    a. **Veća vrednost PSNR-a znači bolji kvalitet slike**, jer ukazuje na to da su izmene neprimetne.

PSNR (dB)	Kvalitet	Vidljivost
> 50	Odličan	Neuočljivo
40-50	Veoma dobar	Teško uočljivo
30-40	Dobar	Može se uočiti pri pažljivoj inspekciji
< 30	Slab	Jasno vidljivo

Ove dve metrike su međusobno povezane – **što je manji MSE, to je veći PSNR**.

MSE daje apsolutnu vrednost greške, dok PSNR izražava tu razliku logaritamski, u odnosu na maksimalnu moguću vrednost piksela.
Zbog toga se PSNR češće koristi za poređenje kvaliteta različitih steganografskih algoritama, jer je intuitivniji i jednostavniji za interpretaciju.

5. ZAKLJUČAK

Ovaj rad je predstavio sveobuhvatnu analizu i implementaciju LSB (Least Significant Bit) steganografije, tehnike koja omogućava skrivanje tajnih poruka unutar digitalnih slika. Kroz teorijski okvir i praktičnu implementaciju, demonstrirano je kako se principi kriptografije i obrade slike mogu kombinovati da bi se postigla efikasna i neuočljiva komunikacija.

LSB steganografija predstavlja balans između kapaciteta za skrivanje podataka i neuočljivosti izmena. Modifikacija samo jednog bita po bajtu (promena vrednosti za maksimalno ±1) rezultira vizuelno identičnim slikama, što potvrđuju PSNR vrednosti redovno iznad 50 dB u našim testovima.
Maksimalni kapacitet slike direktno zavisi od njenih dimenzija. Za standardnu sliku rezolucije 1920×1080 piksela, moguće je sakriti približno 750 KB teksta, što je dovoljno za brojne praktične primene - od tajnih poruka i watermarkinga do embeddinga metadata.

Kritično je da enkodovane slike budu sačuvane u lossless formatima (PNG, BMP). JPEG kompresija koristi aproksimacije boja koje nepovratno menjaju LSB bitove, čineći ekstraktovanje poruke nemogućim. Ova karakteristika predstavlja inherentno ograničenje LSB tehnike u kontekstu modernog web-a gde dominira JPEG format.

Steganografija, kao i kriptografija, je područje gde je lako napraviti nešto što *izgleda* sigurno, ali je zapravo ranjivo. Zato je razumevanje fundamentalnih principa, ograničenja i pretnji ključno za svakoga ko želi da koristi ili razvija steganografske sisteme u realnim scenarijima. Implementiran algoritam je ranjiv na statističke metode detekcije kao što su chi-square test i RS analysis. Sekvencijalna distribucija podataka kroz piksele ostavlja karakterističan statistički potpis koji sofisticirani steganalitički alati mogu detektovati.

6. LITERATURA

1. Fridrich, J. (2009). *Steganography in Digital Media: Principles, Algorithms, and Applications.* Cambridge University Press.
2. Shih, F. Y. (2017). *Digital Watermarking and Steganography: Fundamentals and Techniques* (2nd ed.). CRC Press.
3. Bobby, S., & Kumar, R. (2017). *Fundamentals for Steganography Using Least Significant Bit Algorithm. International Education and Research Journal (IERJ)*
4. Wang, S., Yin, H., & Wang, X. (2022). *Research on the Improvement of LSB-based Image Steganography Algorithm. Advances in Journal of Science and Technology,*