Univerzitet u Nišu
Elektronski fakultet

Implementacija F5 algoritma

Seminarski rad

Digitalna forenzika

Predmetni nastavnik:
Bratislav Predić

Student:
Petar Brajković, br. indeksa 1733

Sadržaj

Uvod

U današnjem digitalnom dobu, gde se informacije brzo šire i dele, potreba za sigurnim prenosom podataka postaje sve važnija. Steganografija, umetnost skrivanja poruka unutar drugih medija, nudi inovativan pristup rešavanju ovog izazova. Ovaj seminarski rad usmerava se na analizu i razumevanje F5 steganografskog algoritma, sofisticirane metode skrivanja informacija unutar digitalnih slika.

Steganografija, izvorno grčki izraz koji znači "sakrivanje pisama", predstavlja skrivanje informacija na način da prisutnost same informacije ostane neprimetna za neovlaštene posmatrače. Za razliku od kriptografije koja se bavi šifriranjem podataka, steganografija fokusira svoju pažnju na prikrivanje činjenice da su podaci prisutni.

Motivacija iza korištenja steganografije često proizlazi iz potrebe za očuvanjem privatnosti, sigurnosti i tajnosti informacija. Steganografski algoritmi nude suptilan način skrivanja podataka unutar digitalnih medija, kao što su slike, zvukovi ili videozapisi, bez ikakvog sumnjičavog izgleda.

Među raznim steganografskim tehnikama, F5 algoritam izdvaja se svojom sposobnošću prilagođavanja slikama te očuvanju kvaliteta i integriteta originalne slike. Ovaj rad usmerava se na razumevanje principa rada F5 algoritma te njegovu primenu u steganografiji s posebnim naglaskom na skrivanje informacija unutar digitalnih slika.

F5 algoritam se smatra relativno složenim, ali istovremeno pruža dobar uvid u osnovne koncepte steganografije. Implementacijom F5 algoritma se postiže bolje razumevanje skrivanja podataka u slikama, što može poslužiti kao dobra osnova za razumevanje drugih steganografskih algoritama.

Steganografija

Steganografija je grana kriptografije koja se bavi skrivanjem informacija unutar drugih ne-sumnjivih podataka. Cilj steganografije je održavanje tajnosti prisustva samih podataka, tj. da se komunikacija ne ometa. Ova disciplina je stara koliko i ljudska istorija, sa mnogim primerima primene kroz vekove.

Tragovi steganografije već su postojali u staroj Grčkoj, kada je Histije, vladar grčkog grada Mileta, poslao tajnu poruku svom savezniku koristeći tetovažu na glavi svog roba.

Tokom srednjeg veka, mnogi su koristili steganografiju kako bi štitili poverljive poruke. Neki od metoda uključivali su skrivanje poruka unutar slika ili teksta, ili čak pisanje poruka ispod drugih poruka. Prvi zabeleženi izraz upotrebio Johanes Tritemije 1499. godine u svojoj „Steganografiji", raspravi o kriptografiji i steganografiji koja je uključivala stvarne steganografske tehnike, prerušenoj u knjigu o magiji. U početku je autor odlučio da ga ne štampa, pa je čak i uništio velike delove, verujući da nikada nije trebalo da ugledaju svetlost dana, ali je tekst nastavio da kruži u vidu privremene verzije i objavljen je posthumno 1606. godine.

Pojavom telegrafije, ljudi su eksperimentisali sa steganografskim tehnikama kako bi skrivali informacije prenesene telegrafskim putem.

Tokom Prvog i Drugog svetskog rata, steganografija je dobila novu dimenziju. Različite zemlje razvijale su složene tehnike za šifrovanje i steganografiju kako bi štitile tajne informacije. Na primer, Morseova azbuka je korišćena za slanje šifrovanih poruka putem radio-talasa.[1]

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.

Steganografija podrazumeva:

- **Skrivanje podataka u slikama:** Modifikacije boje ili nijansi pojedinih piksela tako da ljudsko oko ne primeti promene.
- **Skrivanje podataka u zvuku:** Dodavanje ili modifikacija frekvencija zvuka tako da ne bude primetno ljudskom uhu.
- **Skrivanje podataka u tekstualnim dokumentima:** Upotreba nevidljivih karaktera ili neprimetnih modifikacija teksta.

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.

- **Frequency Domain Techniques:**
  - Ove tehnike uključuju manipulaciju frekvencijama slike ili zvuka kako bi se skrile informacije. Primeri uključuju Fourierovu transformaciju, koja transformiše signal iz vremenskog domena u frekvencijski domen.
- **Transform Domain Techniques:**
  - Algoritmi poput Discrete Cosine Transform (DCT) i Discrete Wavelet Transform (DWT) koriste se za skrivanje informacija u transformiranim domenima slike ili zvuka.
- **Spread Spectrum Technique:**
  - Ova tehnika distribuira informacije kroz široki spektar frekvencija, čime se otežava detekcija. Primena uključuje skrivanje informacija u različitim delovima spektra, poput frekvencija boje u slici.
- **Echo Hiding:**
  - Informacije se ubacuju u eho zvuka, često korišćeno u steganografiji zvučnih datoteka. Tajna poruka se "uklapa" u eho i postaje teško uočljiva.
- **Visual Cryptography:**
  - Ova tehnika se često koristi za deljenje tajnih slika. Originalna slika se deli na delove, a tajna se može videti samo kada se kombinuju određeni delovi.
- **F5 Algorithm:**
  - Specifično dizajniran za skrivanje informacija u slikama, F5 koristi prilagodljivu kvantizaciju kako bi umetnuo podatke na način koji minimalno utiče na kvalitet slike.
- **Digital Watermarking:**
  - Ova tehnika se koristi za trajno obeležavanje digitalnih sadržaja sa skrivenim informacijama, obično u cilju zaštite autorskih prava.

F5 algoritam

F5 (F5 Algorithm) je steganografski algoritam koji je razvijen za skrivanje informacija unutar slika. Ovaj algoritam je posebno dizajniran kako bi se efikasno skrivali podaci unutar kompresovanih slika.

Koristi Least Significant Bit (LSB) Substitution tehniku, gde se najmanje značajni bitovi boja piksela u slici koriste za skrivanje dodatnih informacija. Ova tehnika omogućava umetanje podataka sa minimalnim vizuelnim uticajem na kvalitet slike. F5 uključuje prilagodljivu kvantizaciju, što znači da se prilagođava nivou kvantizacije u zavisnosti od sadržaja slike.

Rezultat prilagodljive kvantizacije je očuvanje kvaliteta slike tokom procesa skrivanja informacija. Algoritam je otporan na različite tehnike detekcije steganografskih informacija. Pokušava izbegavati statističke anomalije koje bi mogle ukazivati na prisustvo skrivenih podataka. Omogućava korisnicima da koriste lozinke i enkripciju kako bi dodatno obezbedili skrivene podatke.

F5 algoritam ima ograničenja u kapacitetu za skrivanje podataka. Umetanje prevelike količine informacija može dovesti do primetnog gubitka kvaliteta slike. F5 algoritam može biti osetljiv na neke transformacije slike, poput kompresije ili promene formata, što može uticati na mogućnost izvlačenja skrivenih podataka.

Aktuelni steganografski algoritmi

Steganografija danas ima značajnu ulogu u oblasti bezbednosti informacija, privatnosti i digitalne forenzike.

Digitalno vodeni žig (digital watermarking) je oblik steganografije koji se često koristi za označavanje digitalnih medija sa informacijama o autorskim pravima, vlasništvu ili izvoru - pomaže u zaštiti intelektualne svojine.

U digitalnoj forenzici, steganografija se često koristi za skrivanje tragova ili informacija koje se koriste u istrazi. Otkrivanje skrivenih podataka postaje ključno u analizi digitalnih dokaza.

Steganografski algoritmi koji se danas koriste:

- **OutGuess:**
  - OutGuess je open source steganografski algoritam koji se često koristi za skrivanje informacija u slikama (posebno prilagođen za JPEG format slika). Koristi metodu najmanjih značajnih bitova u pikselima slike kako bi umetnuo tajne podatke. Ima široku primenu zbog svoje jednostavnosti i efikasnosti.

- **Steganography Toolkit (S-Tools):**
  - S-Tools je kolekcija alata za steganografiju koja uključuje različite tehnike skrivanja podataka u slikama, zvucima i drugim digitalnim medijima.
- **F5 Algorithm**
- **Steghide:**
  - Steghide je open source alat koji omogućava korisnicima da skrivaju podatke u slikama i zvucima. Podržava steganografiju u JPEG slikama i WAV zvučnim datotekama. Pruža mogućnosti enkripcije i lozinke za dodatnu sigurnost.
- **WavSteg:**
  - WavSteg je alat za skrivanje informacija u audio datotekama, posebno WAV formatima. Koristi različite tehnike steganografije kako bi se postigla efikasnost.

Implementacija

Implementacija F5 algoritma biće izvedena u programskom jeziku Python. Biblioteke koje su korišćene u radu:

- PIL (Python Imaging Library) – biblioteka za manipulaciju slikama
- Tkinter - biblioteka za grafički korisnički interfejs (GUI)

Kao dodatak radu, a kao uvod u steganografiju i F5 algoritam, implementiran je kratak algoritam za upis podataka u sliku i iscitavanje podataka iz slike koji je demonstriran u nastavku(Slika 1).

```python
from PIL import Image
from tkinter import Tk, filedialog
STEP=3
def choose_image():
   root= Tk()
   root.withdraw()

   carrier_path=filedialog.askopenfilename(title="Izaberite sliku", filetypes=[("PNG files", "*.png")])
   output_path = input("Unesite naziv slike sa sakrivenim podacima (sa ekstenzijom na kraju): ")

   if carrier_path:
      print("Izabrana slika:",carrier_path)
      data_to_hide=input("Unesite tekst koji zelite da sakrijete:")
      f5_encode(carrier_path, data_to_hide, output_path)
      print(f5_decode(output_path,  len(data_to_hide)))


def f5_encode(carrier_path, data_to_hide, output_path):
 carrier = Image.open(carrier_path).convert('RGB')
 binary_data = [len(data_to_hide)] + [ord(char) for char in data_to_hide]
 pixels = list(carrier.getdata())
 positions = list(range(0, len(pixels), STEP))[:len(binary_data)]

 for i, position in enumerate(positions):
    pixel = list(pixels[position])
    pixel[0] = binary_data[i]
    pixels[position] = tuple(pixel)

 new_image = Image.new('RGB', (carrier.width, carrier.height))
 new_image.putdata(pixels)
 new_image.save(output_path)

def f5_decode(encoded_path, message_length):
    encoded_image = Image.open(encoded_path)
    pixels = list(encoded_image.getdata())
    decoded_data = ''
    pixel=pixels[0]
    message_length = pixel[0]
    for i in range(STEP, message_length*STEP+1,STEP):
        pixel=pixels[i]
        decoded_data+=chr(pixel[0])
    return decoded_data

# Primer korišćenja
choose_image()
```

Slika 1: Skrivanje podataka u sliku i čitanje skrivenih podataka iz nje

Ova verzija algoritma se bazira na tri funkcije:

- **choose_image() -** pokreće celokupni proces skrivanja i otkrivanja teksta unutar slike. Otvara dijalog za odabir slike u koju će se upisati skriveni podaci, zatim se od korisnika zahteva unos putanje za novu sliku sa skrivenim podacima i tekst koji će se sakriti u slici. Nakon unešenih podataka, zovu se, redom, algoritmi za skrivanje i čitanje skrivenih podataka iz slike.
- f5_encode(carrier_path, data_to_hide, output_path) – otvara sliku sa putanje *carrier_path,* podatke koje treba sakriti (*data_to_hide*) konvertuje u ASCII vrednosti i na početak te liste stavlja dužinu podataka koje treba sakriti. Takođe, pamti se niz pozicija u piksela slike na čije će se prve (R komponenta piksela) komponente upisivati skriveni podaci. Pozicije će imati korak *STEP* koji je definisan kao konstanta, na početku programa. Proslaskom kroz vrednosti piksela slike se upisaju podaci, a zatim se od novodobijenih piksela kreira nova slika i čuva na putanji *output_path.* Kao prva vrednost se upisuje dužina podatka koji se sakriva.
- **f5_decode(encoded_path, message_length) –** otvara se slika sa putanje *encoded_path,* čiji se pikseli čitaju, zatim se iz prvog piksela čita dužina sakrivenog teksta. Iteracijom kroz niz piksela sa korakom STEP se čitaju vrednosti prve komponente piksela, konvertuju se u karakter i konkateniraju u poruku koju funkcija vraća.

Treba naglasiti da ovaj kod ima par ograničenja kao što su:

- tekst koji se sakriva nije enkriptovan
- unapred izabran korak između dva piksela koja će imati skrivenu vrednost i upisana vrednost dužine poruke u prvom pikselu.

Složenija verzija algoritma je prikazana na narednoj slici (Slika 2).

```python
from PIL import Image
from tkinter import Tk, filedialog
STEP=3

def get_frequencies(image):
    frequencies = {}
    pixels = list(image.getdata())
    for pixel in pixels:
        key = tuple(pixel)
        frequencies[key] = frequencies.get(key, 0) + 1
    return frequencies

def choose_pixel(frequencies, used_pixels):
    sorted_frequencies = sorted(frequencies.items(), key=lambda x: x[1], reverse=True)
    for pixel, _ in sorted_frequencies:
        if pixel not in used_pixels:
            return pixel

def f5_decode(encoded_path, message_length,positions):

    encoded_image = Image.open(encoded_path)
    pixels = list(encoded_image.getdata())
    decoded_data = ''
    for i in positions:
        pixel=pixels[i]
        decoded_data+=chr(pixel[0])
    return decoded_data

def f5_encode(carrier_path, data_to_hide, output_path):
 carrier = Image.open(carrier_path)
 binary_data =[ord(char) for char in data_to_hide]
 pixels = list(carrier.getdata())
 used_pixels = set()
 new_image = Image.new('RGB', (carrier.width, carrier.height))
 positions =list()
 for i in range(len(binary_data)):
        pixel = choose_pixel(get_frequencies(carrier), used_pixels)
        used_pixels.add(pixel)
        position = pixels.index(pixel)
        positions.append(position)
        pixel = list(pixel)
        pixel[0] = binary_data[i]
        pixels[position]=tuple(pixel)
 new_image.putdata(pixels)
 new_image.save(output_path)
 return positions

def choose_image():
   root= Tk()
   root.withdraw()
   carrier_path=filedialog.askopenfilename(title="Izaberite sliku", filetypes=[("Slike","*.png")])
   output_path = input("Unesite naziv slike sa sakrivenim podacima:")
   if carrier_path:
      print("Izabrana slika:",carrier_path)
      data_to_hide=input("Unesite tekst koji zelite da sakrijete:")
      positions=f5_encode(carrier_path, data_to_hide, output_path)
      print(f5_decode(output_path,  len(data_to_hide),positions))

choose_image()
```

Slika 2: Implementacija F5 algoritma

Dodatak ovom algoritmu su funkcija za određivanje frekvencije pojavljivanja svakog piksela u slici i funkcija koja bira naredni piksel u koji će se upisati podaci.

- get_frequencies(image) – vraća key value parove, gde je ključ vrednost piksela, a vrednost njegovo pojavljivanje u slici.
- choose_pixel(frequencies, used_pixels) – prima key value parove piksela, bira naredni piksel u koji će se upisati podatak, a koji nije već korišćen.

Korišćenje ovih funkcija je donelo i oderđene izmene u fukciji *f5_encode.* Naredni piksel se bira pomoću funkcije *choose_pixel,* koji se upisuje u niz *used_pixels* kako se ne bi upisivalo u iste piksele.

Ovaj algoritam je poboljšanje prethodnog jer bira piksele na osnovu frekvencije pojavljivanja kako bi smanjio uticaj na kvalitet slike i povećao kapacitet za sakrivanje podataka.

Literatura

[1] Steganography: from its origins to the present - Telsy

https://www.freecodecamp.org/news/what-is-steganography-hide-data-inside-data/

https://www.comptia.org/blog/what-is-steganography

Digitalna forenzika