УНИВЕРЗИТЕТ У НИШУ

ЕЛЕКТРОНСКИ ФАКУЛТЕТ

Стеганографија на примеру Spread Spectrum технике

Семинарски рад

Студијски програм: Рачунарство и информатика

Модул: Софтверско инжењерство

Ментор:

Проф. др Братислав Предић

Студент:

Лазар Милосављевић 1768

Ниш, Септембар 2024.

Садржај

Увод

Стеганографија је научна дисциплина која се бави прикривеном разменом информација. Реч стеганографија изведена је од грчких речи стеганос и графеин, што у дословном преводу значи "скривено писање". Основни принцип стеганографије почива на прикривању самог постојања информације која се преноси унутар неког наизглед безопасног медија или скупа података. Модерна стеганографија, која користи предности дигиталне технологије, најчешће подразумева скривање тајне поруке унутар неке мултимедијалне датотеке, нпр. слике, аудио или видео датотеке. Мултимедијалне датотеке по правилу садрже неупотребљене или неважне просторе података које различите стеганографске технике користе тако да их попуне са тајним информацијама. Такве датотеке се потом могу размењивати без да ико буде свестан праве сврхе дотичне комуникације.

Стеганографија има веома широке могућности примене - од прикривене размене података у приватне и пословне сврхе па све до заштите ауторских права у облику воденог жига. Но, због свог темељног принципа "невидљивости" информација, често се користи и током нелегалних активности. За разлику од шифровања, које спроводи комплексне трансформације података како би их учинило нечитљивим, стеганографија представља технику скривања тајних порука на такав начин да нико, изузев пошиљаоца и примаоца, коме је порука намењена, не може ни да наслути постојање поруке. Медијум на коме је порука скривена може бити јавно доступан свима, али само неко ко зна за постојање скривене поруке ће применити одговарајућу технику да је издвоји.

Стеганографија

Историја стеганографије

Стеганографија датира још од 200. год. пре Христа, иако не у облику у каквом нам је данас позната. Један од таквих примера су Наска линије у Перуу. На сувој равни је изгребано више од 80 км линије између два града Наска и Палпе, од којих су многе видљиве тек из ваздуха.

Трагови стеганографије могу се пронаћи у делима Џона Тритемијуса (1462-1516), који је био немачког порекла („Стеганографија: уметност која захтева откривање скривеног писања мисаоним активностима човека“). Краљица Марија од Шкотске је користила комбинацију криптографије и стеганографије да сакрије писма. Она је своја писма скривала у рупи за запушавање бурета пива, које је неометано улазило и излазило из њеног затвора. Херодот је у 5. веку пре нове ере бријао главе носиоцима порука, писао је охрабрујуће поруке Аристагорасу из Милета, које су подстицале на револт против краља Персије. Након што би гласнику порасла коса, он би био послат примаоцу. Када би гласник стигао на предодређено одредиште, глава би му поново била обријана и на тај начин би се дошло до скривене поруке.

480. године пре нове ере Грк по имену Демарат послао је Спартанцима поруку упозорења о неизбежној инвазији Ксеркса. Херодот описује метод који је користио Демарат: „Како је опасност од откривања била велика, постојао је само један начин на који је могао да пошаље поруку: тако што је са дрвене табле састружан восак, затим исписивана на њој порука шта Ксеркс планира да уради, а онда је порука сакривана поново воском. На овај начин табле, наизглед чисте, не би проузроковале проблеме са стражарима дуж пута.“

У новијој историји, неке од стеганографских метода коришћене су током Другог светског рата. Микротачке развијене од стране нациста биле су делови микрофилма, направљени под великим увећањем, најчешће преко 200 пута. Ове тачке могле су да садрже странице информација, слика итд. Нацисти су такође користили невидљиво мастило и безшифарну технику. Безшифарна техника (или кодирање) се користила да се сакрије једна порука у другу без коришћења компликованог алгоритма. Још један од примера стеганографије укључује употребу Карданове решетке.

Научни приступ изучавању стеганографије практично је започет 1983. До 11. септембра 2001. године, стеганографија се сматрала безопасном. Одједном, после више од две хиљаде година постојања, постала је предмет општег интересовања. За њену употребу заинтересовала се и влада најразвијеније земље света – САД. Због могућности опште употребе, својства маскирања и могућности коришћења у области индустријске шпијунаже, стеганографија је постала интересантна и за пословни свет.

Терористи Ал Каиде организовали су и извели напад на Светски трговински центар и Пентагон комуницирајући путем неколико веб сајтова. План терористичког напада, у облику шаљивих коментара, био је „утискнут“ у слике које су преношене путем интернета. Иако наизглед застарео и наиван, стеганографски метод преноса порука

показао се крајње ефикасним. Систем безбедности САД био је потпуно немоћан. Терористи су надмудрили експерте и америчке безбедносне агенције.

Савремена стеганографија

Савремени појам стеганографије, односно стеганографског система (стегосистема) подразумева скуп средстава и метода који се користе за формирање скривеног канала преноса информација. Општи процес стеганографије (слика 1.) дат је релацијом:

Стеганографски медијум = скривена порука + носилац поруке + стеганографски кључ

Ова релација описује како се стеганографски медијум формира комбиновањем тајне поруке, медијума који је носи, и стего кључа који осигурава тајност и сигурност поруке током преноса. Стеганографски медијум је резултат овог процеса и представља објекат који садржи скривену информацију, али изгледа као обичан, безазлени податак.

У својству података може се користити било која информација (текст, аудио подаци или слика), па је боље користити термин порука (енгл. *message*) уместо информација. Носилац поруке (енгл. *carrier, cover, cover medium*) је било која информација намењена да као „носилац“ пренесе скривену поруку.

Уграђена или тајна порука (енгл. *embedded message*) је порука која се имплементира у носиоца поруке.

Стего кључ (енгл. *steg-key*) је тајни кључ помоћу кога се тајна порука имплементира у носиоца поруке. У стегосистему може постојати један или више стего кључева, а по аналогији са криптографијом, разликујемо стего системе са тајним и јавним кључем. Стеганографски медијум (енгл. *steganography medium, stego-medium*) је посредник који садржи имплементирану поруку која се тајно преноси.

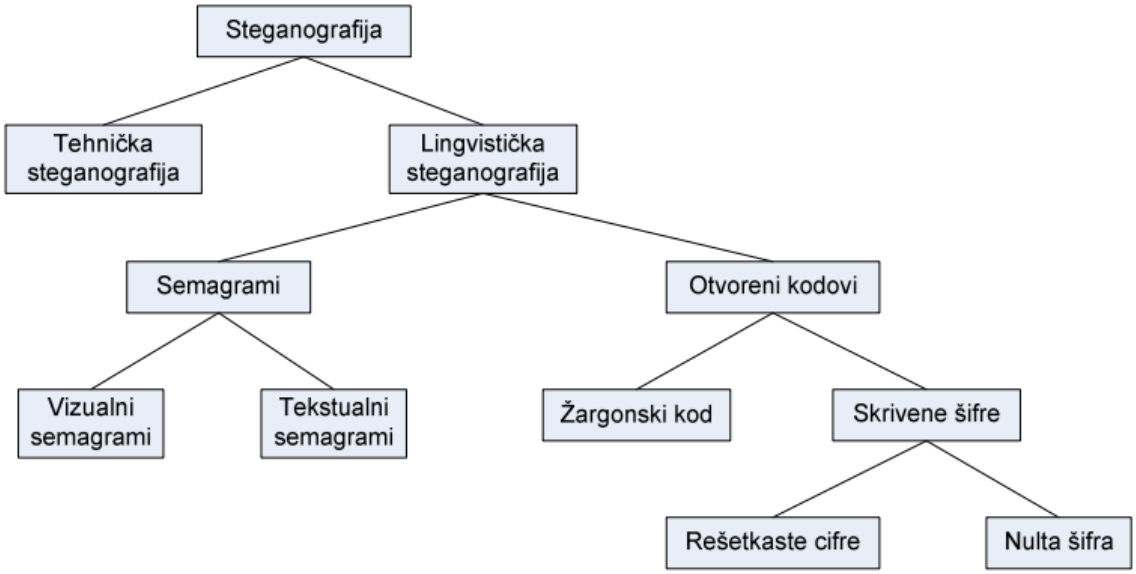
Стеганографски канал (стегоканал) је комуникациони канал преко којег се шаље стего носилац поруке.

Примена стеганографије се најчешће базира на следећем принципу: пошиљалац тајне поруке насумично бира носиоца поруке; у изабрани носилац поруке имплементира се тајна порука уз помоћ стего кључа; примаоцу се шаље стеганографски медијум, а прималац на другој страни обрнутим поступком долази до садржаја тајне поруке. Да тајна порука не би била видљива, битно је да носилац поруке садржи довољно редундантних битова који могу бити замењени тајном поруком.

Важно је напоменути да нису сви дигитални формати погодни за преношење тајних порука, јер се, на пример, променом битова у неком извршном фајлу доводи до тога да програм не ради или јавља грешке при раду.

Класификација стеганографских техника

На слици 1. представљена је подела стеганографских техника.



слика 1 – подела стеганографских техника

Техничка стеганографија користи научне методе за скривање порука, као што је употреба невидљивог мастила или микротачака и друге методе смањења величине тајних порука.

Лингвистичка стеганографија крије поруке у носиоце на неки неуобичајен начин и даље их категорише као семаграме или отворене кодове. Семаграми крију информације кроз симболе или знакове. Визуелни семаграми користе наизглед безопасне или свакодневне физичке облике за пренос порука, као што су постављање детаља на десктопу или веб сајту. Текстуални семаграми крију поруке модификовањем изгледа носиоца текста, као што је вешта промена типа или величине слова, додавање додатних празних места, улепшавање слова или додавање ручно писаног текста.

Отворен код крије поруке у легитимне носиоце порука на начин који није очигледан некоме ко није сумњичав посматрач. Носилац поруке се понекад назива јавно саопштење, с обзиром на то да је скривена порука заправо прикривено саопштење. Ова категорија је подељена на жаргонски код и сакривени код.

Жаргонски код, као што му име сугерише, користи језик који је разумљив само одређеној групи људи, док је за друге безначајан. Жаргонски код може укључивати симболе који указују на присуство и врсту сигнала бежичне мреже, подземну терминологију или, наизглед безазлену, конверзацију која преноси посебно значење познато само говорнику. Подскуп жаргонског кода је **знаковни код**, где се одређеним преуређењем фразе мења њено значење.

Сакривени (тајни) код крије поруку јавно у носећој поруци, тако да свако ко зна метод скривања може да је открије. Пример је **решеткасти код**, који користи шему (шаблон) за откривање скривене поруке у носиоцу. Када се шаблон примени на текст, појављује се сакривена порука.

Безшифарна техника крије поруку применом одређеног скупа правила, попут „прочитај сваку пету реч“ или „погледај свако треће слово у свакој речи“.

Компјутерска стеганографија

Да би се сакрила порука у дигиталном фајлу користећи безшифарну технику, нису потребни специјални алати или посебне вештине. На пример, слика или текстуални блок могу бити сакривени у другој слици у PowerPoint фајлу. Порука се такође може сакрити у атрибутима Word фајла, у коментарима на веб страници или у формату који веб претраживач игнорише. Текст може бити сакривен као стилизована линија у документу, тако што му се додели боја позадине, а затим постави на други цртеж у првом плану. Прималац може да поврати текст променом боје текста. Иако су ови механизми технички једноставни, често су веома ефективни.

Компјутерска стеганографија је део стеганографије који се бави применом стегосистема уз помоћ компјутерске технике. Савремени методи компјутерске стеганографије заснивају се на:

1. Коришћењу специјалних својстава компјутерских формата.
2. Статистичком преобиљу информација у аудио и видео дигиталним сигналима.

Специјална својства компјутерских формата мање се користе за заштиту информација, а више за нападе, као што је убацивање злонамерних програма. На пример, вирус W32/Perrun се убацује у .jpg фајлове, формирајући двокомпонентни програм.

Претек информација у аудио и видео дигиталним сигналима представља главни правац развоја метода компјутерске стеганографије. Дигиталне фотографије, музика и видео садрже низове бројева који кодирају интензитет светлости (за фотографију) или звука (за музику) у одређеном тренутку. Битови са најмањим значењем садрже мало корисних информација, па њихова замена тајним порукама не утиче значајно на квалитет. Како дигитални сигнали већ садрже квантни шум, ова замена не изазива уочљиво погоршање квалитета.

Методи у овој области се деле на:

1. **Метод заснован на замени најмање значајних битова** (LSB - least significant bits)
2. **Метод заснован на трансформацији слике**, попут дискретне косинусне трансформације (DCT - Discrete Cosine Transformation) или таласних трансформација (Wavelet Transformation).

Стеганографија у видео форматима

Када се информација крије унутар видео фајла, програм или особа обично користе метод дискретне косинусне трансформације (DCT), који незначајно мења сваку слику у видеу, чинећи промене невидљивим људском оку. DCT модификује вредности одређених делова слике, обично их заокружујући. На пример, ако део слике има вредност 6.667, она ће бити заокружена на 7.

Стеганографија у видео фајловима је слична стеганографији у сликама, али информација се крије у сваком фрејму видеа. Када се у видео уметне само мали део информације, то генерално није уочљиво. Што више информација се сакрије, постаје мање уочљиво, али и ризик од откривања расте. Ова метода омогућава сигурну преношење порука или података без значајног утицаја на квалитет видеа, што је идеално за разне примене, од безбедности до приватне комуникације.

Стеганографија у аудио форматима

Када се информација крије у аудио фајлу, обично се користи техника кодирања на слабом биту (LSB - least significant bit), која има сличности са LSB методом у сликама. Међутим, проблем кодирања на слабом биту у аудио формату је уочљивост промене на звучном нивоу, што чини овај метод донекле ризичним.

Метод проширеног спектра (Spread Spectrum) за сакривање информација у аудио фајлу функционише тако што се додају случајни шумови у информације које су сакривене у носиоцу, распоређујући их кроз спектар фреквенција. Ова техника помаже у минимизовању уочљивости скривених података, што их чини теже детектовати.

Метод сакривања података уз помоћ еха (Echo Data Hiding) користи ехо у музичким фајловима за скривање информација. У овом методу, информација се може сакрити једноставним додавањем додатног звука у ехо у аудио фајлу. Овај метод је често бољи од других, јер може побољшати квалитет звука аудио фајла, чинећи сакривену информацију мање уочљивом и задржавајући високу репродукцију звука.

Ове технике пружају разне могућности за сигурно преношење информација у аудио фајловима, уз минимални утицај на квалитет звука.

Стеганографија у документима

Коришћење стеганографије у документима функционише једноставним додавањем празних места и табова на крајевима редова у документима. Овај вид стеганографије је веома ефикасан, јер употреба празних места и табова није видљива људском оку, бар у највећем броју текстуалних уредника.

Празна места и табови су природно присутни у документима, што значи да не постоји никакав могући начин да употреба ове методе стеганографије буде некоме сумњива. На тај начин, поруке могу бити скривене у тексту без остављања видљивих трагова, што ову технику чини атрактивном за тајну комуникацију и заштиту информација.

Стеганографија у сликама

За скривање информација у сликама најчешће се користи метод **LSB** (Least Significant Bit). За компјутер слика представља једноставан фајл који показује различите боје и интензитете светлости у различитим деловима слике. Најбољи формат слике за скривање информација је **24-битна BMP слика**, као највећи тип фајла, а самим тим и најквалитетнији. Много је лакше сакрити информације у слици високог квалитета и резолуције. У BMP формату, боја је представљена као комбинација црвене, зелене и плаве боје. Сваку од ових боја представља 1 бајт. Комбинација од по једног бајта црвене, зелене и плаве даје 3 бајта (24 бита) који представљају 1 пиксел. 24-битна шема подржава 16.777.216 (2^24) јединствених боја. Иако је 24-битна слика најбоља за скривање информација, због њене величине многи се опредељују за 8-битни BMP, или нпр. GIF. Један од разлога је тај што слање великих слика преко Интернета може изазвати сумњу.

Највећи број апликација за дигиталне слике данас подржава 24-битне слике, где је сваки пиксел кодирано у 24 бита. Друге апликације кодирају боје користећи 8 бита/пикселу. Свака пиксел је кодирана са 8 бита где вредност тачке у 24-бита боје улази у палету. Овај метод ограничава јединствени број боја у датој слици на 256 (2^8). Избор кодирања боје свакако утиче на величину слике. Слика димензија 640 x 480 пиксела користећи 8-битни сликовни формат заузима отприлке 307 KB (640x480 = 307.200 бајта), док би слика димензија 1400x1050 користећи 24-битни сликовни формат заузела 4.4 MB (1400x1050x3 = 4.410.000 бајта). Палета боја и 8-битни сликовни формат су обично коришћени у **GIF** (Graphics Interchange Format) и **BMP** формату. GIF и BMP се генерално узимају у обзир да би се обезбедила идеална компресија зато што обновљена слика после кодирања и компресије је идентична у бит оригиналној слици.

JPEG (Joint Photographic Experts Group) формат слике користи **DCT** (Discrete Cosine Transform), пре него кодирање пиксел-по-пиксел. У JPEG формату, слика је подељена на 8x8 блокова за сваку одвојену компоненту боје. Циљ је пронаћи блок где би промена вредности пиксела била најмања. Ако је вредност превелика, блок се дели на 8x8 подблокова, све док вредност не буде довољно ниска. Свака 8x8 блок, или подблок, је трансформисан у 64 DCT коефицијента који су приближни осветљењу (светлости, тами и контрасту) и сигналу боје тог дела слике. JPEG се генерално сматра лошим форматом за компресовање, јер слика обновљена из компресије је блиска оригиналној слици, али није и идентична. Да би ставили што већу количину података у слику, користе се две врсте компресије података: без губитака (Lossless) и са губицима (Lossy) компресија. Прави представник lossless компресије су .ZIP и .RAR, а lossy компресије - JPEG. За компресију података може се користити било који алгоритам.

За поправку мутњикавости, шаренило и „усамљене тачке“ на слици, могу се користити различити, пажљиво одабрани филтери и ефекти за обраду слике. Маскирање и филтрирање је једна од најзаступљенијих техника у стеганографији. Ако се примети у

неком делу слике мутњикавост после убацивања скривених података, онда се оригинал слике може мало потамнити и након тога убачени подаци. Успех се заснива на чињеници да људско око слабије распознаје разлике међу бојама слабије сјајности. Додавање „шума“ је једна од веома ефектних техника. Заснива се на додавању белих тачака или избељивању неких тамнијих. Све ово крајњем кориснику ствара утисак да је слика скенирана и да је слабијег квалитета и такву слику ретко ко узима у детаљно разматрање. При додавању шума, важно је да се не мењају битови који носе скривени податак. Корекција боје пиксела је, до сад, најсавршенија техника. Функционише тако што се скривени податак уписује нпр. у сваки други пиксел, а пиксели који остану слободни, коригују се према бојама суседних пиксела. Корекција слободног пиксела се врши тако што се, поред његове оригиналне боје, гледају и оне суседне (слика боље изгледа ако се гледа неколико суседних), а потом се нађе средња вредност боје. Тако ће слика изгледати скоро исто као и оригинал, односно приближније него пре. Недостатак технике је што се користи сваки други пиксел за скривање података, па има двоструко мање простора за податке. Велика предност је то што се шансе за откривање тајног садржаја своде на минимум. Слика ће још боље изгледати ако се подаци скривају у сваком трећем пикселу итд., али ће бити сразмерно мање места. Корекција боје пиксела може се користити на сликама великог формата за скривање података мале дужине.

Spread Spectrum техника у стеганографији

Spread spectrum техника (техника проширеног спектра) представља метод стеганографије и комуникације који омогућава скривање информација у сигналима, укључујући и слике, на начин који их чини тешким за детекцију. Ова техника је заснована на расподели података преко широког спектра фреквенција, што помаже у смањењу вероватноће откривања скривених информација. Спред спектрум методе се користе у различитим апликацијама, од безбедне комуникације до стеганографије.

Основни принципи spread spectrum технике

1. **Дистрибуција сигнала**: Спред спектрум технике користе технику која дели сигнал на много нижих фреквенција. Ова подела омогућава да подаци буду распоређени преко великог опсега фреквенција, чиме се смањује густина података на свакој појединачној фреквенцији.
2. **Системи за модулацију**: У спред спектрум комуникацијама, подаци се модулишу на начин који омогућава да се шири спектар сигнала. Најчешће коришћени методи модулације су **Direct Sequence Spread Spectrum (DSSS)** и **Frequency Hopping Spread Spectrum (FHSS)**.
    - **Direct Sequence Spread Spectrum (DSSS)**: Ова метода подразумева модулацију података са високом брзином коришћењем низа бита, познатог као *спред код*, који проширује оригиналне податке на много већи опсег фреквенција. Ово помаже у спречавању сметњи и повећава безбедност, јер се оригинални подаци чине нечитљивим без познавања спред кода.
    - **Frequency Hopping Spread Spectrum (FHSS)**: Ова метода подразумева промену фреквенције сигнала на којој се подаци преносе у временским интервалима. Овај метод смањује ризик од откривања и сметњи, јер

нападач мора да зна редослед промене фреквенција да би могао да декодира сигнал.

3. **Смањење сигнала**: Пошто се подаци распоређују на ширем спектру, сигнал је мање подложан сметњама и ометању. Чак и ако неки делови сигнала буду изгубљени или ометени, остатак информација и даље може бити успешан за декодирање.

Примена spread spectrum технике

1. **Бежичне комуникације**: Спред спектрум се широко користи у бежичним комуникацијама, укључујући Wi-Fi, Bluetooth и GSM. Ове технологије омогућавају сигурнију комуникацију и смањују могућност сметњи.
2. **Војна и безбедносна комуникација**: У војним апликацијама, spread spectrum технике се користе за комуникацију у условима високе опасности, где би традиционални сигнали могли бити лако откривени или ометани.
3. **Скривање информација**: У стеганографији, spread spectrum технике омогућава скривање информација у сликама или звуковима. На пример, информације се могу распоредити по пикселима слике на начин који минимизује визуелне промене.

Предности spread spectrum технике

- **Сигурност**: Пошто се подаци распоређују преко великог спектра, тешко их је открити и декодирати без одговарајуће опреме и знања.
- **Отпорност на сметње**: Ова метода чини сигнал мање подложним сметњама, што побољшава квалитет комуникације.
- **Флексибилност**: Спред спектрум технике се могу прилагодити различитим апликацијама и захтевима.

Недостатци spread spectum технике

- **Сложеност**: Имплементација spread spectrum спектрум система може бити сложена и захтевати напредне техничке вештине.
- **Потреба за ресурсима**: Ова техника може захтевати више ресурса у поређењу са традиционалним методама комуникације, као што су пропусност и обрада.

Апликација

Апликација је направљена тако да омогућава скривање поруке унутар слике при чему се користи spread spectrum техника за имплементирање стеганографије. Апликација омогућава генерисање нове слике у којој је скривена порука и коју је могуће декодирати коришћењем истог кључа као и приликом кодирања при чему ће порука бити приказана на екрану.

Кратак опис коришћених технологија и библиотека

Апликација је имплементирана коришћењем програмског језика Python. Имплементација користи неколико важних технологија и Python библиотека. Оне служе за рад са сликама, манипулацију подацима и развој графичког корисничког интерфејса (GUI). Пре свега то с библиотеке: Pillow, NumPy, tkinter, random, os.

Pillow (PIL - Python Imaging Library)

Pillow је библиотека за рад са сликама у Python-у, која је унапређена верзија старије библиотеке PIL. Омогућава учитавање, модификовање и чување слика у разним форматима као што су PNG, JPEG, BMP, и други.

У овој имплементацији, Pillow се користи за отварање слика, претварање слика у низове пиксела и поновно чување слика након уметања тајних података. Функције као што су `Image.open` и `Image.fromarray` омогућавају манипулацију сликама.

NumPy

NumPy је популарна библиотека за научно рачунање у Python-у која пружа подршку за рад са векторима и матрицама, као и великим низовима података. Пружа једноставне и ефикасне алате за обраду и анализу великих сетова података.

NumPy се у овој имлементацији користи за претварање слике у матрицу података (низ пиксела). Ово омогућава приступ сваком пикселу у слици и манипулацију вредностима боја у сваком пикселу. Слика се конвертује у NumPy низ користећи `np.array(image)`, што омогућава да се битови поруке уметну у пикселе слике.

tkinter

`tkinter` је Python-ова уграђена библиотека за креирање графичких корисничких интерфејса (GUI). Омогућава израду прозора, дугмади, улазних поља и других компоненти које корисник може да види и са којима може да интерагује.

Ова библиотека је коришћена за креирање интерфејса апликације. Постоје два таба: један за кодирање (уметање поруке у слику) и један за декодирање (извлачење поруке из слике). Корисник бира слику, уноси тајну поруку или кључ (seed), а затим апликација врши операције кодирања или декодирања.

random

`random` је Python-ова библиотека која омогућава генерисање псевдослучајних бројева. Може се користити за мешање редоследа елемената или одабир насумичних бројева из одређеног опсега.

У овом коду, `random` се користи за генерисање насумичних позиција пиксела у слици на основу задатог "seed"-а. Ова функција омогућава да се порука уметне на насумичне локације у слици, што повећава сигурност и прикривање поруке.

os

`os` је Python-ова библиотека која пружа алате за рад са оперативним системом, као што су рад са фајл системом, креирање, брисање или модификовање датотека.

У овој имплементацији, `os` је коришћен за добијање путање и екстензије фајла, као и за креирање новог фајла са уметнутом поруком. На пример, слика са уметнутом поруком се чува под новим именом које укључује "_encoded".

Имплементација

```python
def generate_random_positions(seed, num_bits, height, width):
    random.seed(seed)
    all_positions = [(x, y) for x in range(width) for y in range(height)]
    random.shuffle(all_positions)
    return all_positions[:num_bits]
```

слика 2 – функција generate_random_positions

Функција `generate_random_positions` служи за генерисање псеудослучајних позиција унутар слике, које ће се користити за уметање (кодирање) или читање (декодирање) тајне поруке. Ова функција је кључна за стеганографију, јер одабир случајних пиксела осигурава да уметање података није предвидљиво.

Функција прихвата четири аргумента:

- `seed`: број који се користи као "кључ" за генерисање истих случајних позиција сваки пут када се овај број користи.
- `num_bits`: број битова поруке које треба уметнути или читати из слике.
- `height`: висина слике (број пиксела по вертикали).
- `width`: ширина слике (број пиксела по хоризонтали).

Функција узима "seed", висину и ширину слике, генерише све могуће позиције пиксела, затим их насумично распореди и враћа првих `num_bits` позиција. Ове позиције ће се користити за уметање или читање битова поруке у стеганографској техници.

```python
def encode(image_path, secret_message, seed):
    image = Image.open(image_path)
    data = np.array(image)

    binary_message = ''.join(format(ord(char), '08b') for char in secret_message)
    binary_message += '1111111111111110'  # Marker kraja poruke

    if len(binary_message) > data.size:
        raise ValueError("Poruka je predugačka za ovu sliku.")

    height, width, _ = data.shape

    positions = generate_random_positions(seed, len(binary_message), height, width)

    for idx, (x, y) in enumerate(positions):
        if idx < len(binary_message):
            bit = int(binary_message[idx])
            data[y][x][0] = (data[y][x][0] & ~1) | bit

    base_name, _ = os.path.splitext(image_path)
    output_path = f"{base_name}_encoded.png"
    encoded_image = Image.fromarray(data)
    encoded_image.save(output_path)
    messagebox.showinfo("Uspeh", f"Poruka je uspešno kodirana u sliku: {output_path}")
```

слика 3 – функција encode

Функција `encode` служи за кодирање (уметање) тајне поруке у слику користећи псеудонасумично одабране позиције пиксела у слици, што је основа стеганографије. Ево детаљног објашњења сваке линије:

Функција `encode` прихвата три аргумента:

- `image_path`: путања до слике у коју желимо да уметнемо поруку.
- `secret_message`: порука коју желимо да кодирамо (уметнемо) у слику.
- `seed`: број који служи као "кључ" за псеудослучајно генерисање позиција пиксела.

Прво се отвара слика коришћењем библиотеке `PIL` (Python Imaging Library). Ова слика ће се користити за кодирање поруке у њене пикселе. Затим се врши конверзија слике у Numpy низ, где сваки елемент представља један пиксел слике. Ово омогућава да се лако манипулише подацима пиксела и убацује се порука у црвени канал сваког пиксела. Након тога конвертује се тајна порука из текста у бинарни формат (нуле и јединице). Сваки карактер у поруци се претвара у свој ASCII код (користећи `ord`), а затим у бинарни низ од 8 битова (користећи `format(..., '08b')`). Резултат је један дугачак бинарни низ који представља целокупну поруку. Додаје се специјални бинарни маркер `'1111111111111110'` на крају поруке. Овај маркер ће омогућити декодирајућој функцији да зна када је порука завршена. Први услов проверава да ли дужина бинарне поруке (укључујући маркер) превазилази број пиксела у слици. Ако је порука предугачка да би

стала у слику, баца се грешка. Затим се преузимају димензије слике из Numpy низа. `height` је висина слике, `width` је ширина слике, а трећи елемент је број канала боје (обично 3 за RGB слике). Након тога ппозива се фунцкија generate_random_positions које је објашњена у претходном делу како би се генерисала листа псеудослучајних позиција пиксела где ће се уметати битови поруке. У наставку се у петљи пролази кроз све псеудослучајне пиксела. Ако је број битова у поруци мањи од укупног броја позиција, престају да се убацују битови. Узима се одговарајући бит из бинарне поруке на датом индексу `idx`, убацује се бит у црвени канал пиксела на позицији (x, y).

- `data[y][x][0] & ~1` се користи да се очисти најмање значајни бит (LSB) црвеног канала.
- `| bit` додаје нови бит (0 или 1) у тај LSB.

Након тога се добија основно име оригиналне слике без екстензије и креира ново име за кодирани фајл. Нови фајл добија суфикс `_encoded` и чува се у PNG формату. Конвертује се модификовани Numpy низ назад у слику користећи библиотеку `PIL`. Ово је кодирана верзија оригиналне слике. Чува се слика и приказује порука о успешном кодирању у графичком интерфејсу (GUI) помоћу прозора за обавештење.

```python
def decode(image_path, seed):
    print("Pokreće se dekodiranje...")
    image = Image.open(image_path)
    data = np.array(image)

    binary_message = ""

    height, width, _ = data.shape
    max_bits = data.size
    positions = generate_random_positions(seed, max_bits, height, width)


    for idx, (x, y) in enumerate(positions):
        if y < height and x < width:
            bit = str(data[y][x][0] & 1)
            binary_message += bit

            if len(binary_message) >= 16 and binary_message[-16:] == '1111111111111110':
                print("Kraj poruke detektovan!")
                break

    secret_message = ""
    for i in range(0, len(binary_message) - 16, 8):
        byte = binary_message[i:i + 8]
        secret_message += chr(int(byte, 2))

    if secret_message:
        messagebox.showinfo("Dekodirana poruka", f"Dekodirana poruka: {secret_message}")
    else:
        messagebox.showinfo("Dekodirana poruka", "Nema dekodirane poruke.")
    return secret_message
```

слика 4 – фунцкија decode

Функција `decode` служи за декодирање (читање) тајне поруке из слике која је претходно кодирана функцијом `encode`.

Функција `decode` која прихвата два аргумента:

- `image_path`: путања до слике из које се извлачи тајна порука.
- `seed`: број који је коришћен за генерисање псеудослучајних позиција пиксела током кодирања.

Отвара слику користећи библиотеку `PIL`. Ово је слика из које треба да се извуче порука, онда се конвертује у Numpy низ, где сваки елемент представља један пиксел слике. Ово омогућава да лако приступимо појединачним пикселима ради читања битова поруке. Димензије слике из Numpy низа чувају се у променљивама `height` и `width`, а трећи елемент се односи на број бојних канала (обично 3 за RGB слике). `max_bits` представља максималан број пиксела у слици, који је једнак укупној величини низа. Позивом функције `generate_random_positions` генерише се листа псеудослучајних позиција пиксела коришћењем истог `seed` броја као током кодирања. Ове позиције ће се користити да би се идентификовало где су битови поруке сакривени у слици.

У петљи се пролази кроз све псеудослучајне позиције пиксела у низу `positions`. Свакој позицији су додељене координате `(x, y)` које представљају позицију у слици. Проверава да ли су координате `(x, y)` у оквиру граница слике (то јест, да ли се пиксел налази у валидној позицији). Извлачи најмање значајни бит (LSB) из црвеног канала пиксела на позицији `(x, y)` користећи операцију `& 1`. Овај бит је део бинарне поруке, јер су битови поруке кодирани битове у LSB током кодирања. Затим се додаје екстраховани бит у стринг `binary_message`, који акумулира све битове тајне поруке. Проверава с да ли је порука достигла дужину од најмање 16 битова и да ли последњих 16 битова у бинарној поруци одговара маркеру за крај поруке `'1111111111111110'`. Ако је маркер детектован, зауставља се петља јер је читање поруке завршено. `secret_message` се користи за чување декодиране тајне поруке у текстуалном формату. Започиње се нова петља која пролази кроз бинарну поруку у блоковима од по 8 битова (сваки блок одговара једном карактеру). Ова петља иде само до претпоследњих 16 битова, јер су последњи 16 бита маркер краја поруке. Узима се један блок од 8 битова из бинарне поруке, који представља један бајт (један карактер поруке). Конвертује се овај блок од 8 битова (који је сада бајт) у одговарајући ASCII карактер користећи функцију `chr` и додаје га у стринг `secret_message`. Проверава се да ли је порука успешно декодирана. Ако јесте, приказује се порука о успеху са декодираним текстом у дијалогу. Ако нема декодиране поруке, приказује се одговарајућа порука и враћа се декодирана порука у текстуалном формату као резултат функције.

```python
def select_image(entry_field):
    file_path = filedialog.askopenfilename(filetypes=[("Image Files", "*.png;*.jpg;*.jpeg")])
    if file_path:
        entry_field.delete(0, tk.END)
        entry_field.insert(0, file_path)
```

слика 5 – функција select_image

Ова функција служи за одабир слике и унос путање до слике у одређено уносно поље у GUI-ју.

Прво се отвара дијалог прозор за одабир фајла који прихвата само слике са екстензијама `.png`, `.jpg`, и `.jpeg`. Путања до изабраног фајла се чува у променљивој `file_path`. Затим се проверава да ли је нека путања изабрана. Ако јесте, чисти се садржај поља за унос (`entry_field`) и у њега убацује путања до изабране слике.

```python
def encode_message():
    image_path = entry_image_path_encode.get()
    secret_message = entry_secret_message.get()
    seed = entry_seed_encode.get()

    if not image_path or not seed.isdigit():
        messagebox.showerror("Greška", "Molimo vas da izaberete sliku i unesete validan seed.")
        return

    encode(image_path, secret_message, int(seed))

def decode_message():
    image_path = entry_image_path_decode.get()
    seed = entry_seed_decode.get()

    if not image_path or not seed.isdigit():
        messagebox.showerror("Greška", "Molimo vas da izaberete sliku i unesete validan seed.")
        return

    decode(image_path, int(seed))
```

слика 6 – функције encode_message и decode_message

Обе функције узимају путању до слике и семе (seed) из одговарајућих уносних поља у GUI-ју. Такође обе функције проверавају да ли је путања до слике унесена и да ли је семе валидно, тј. број. Ако неки од ових услова није испуњен, приказује се порука о грешци и функција се зауставља.

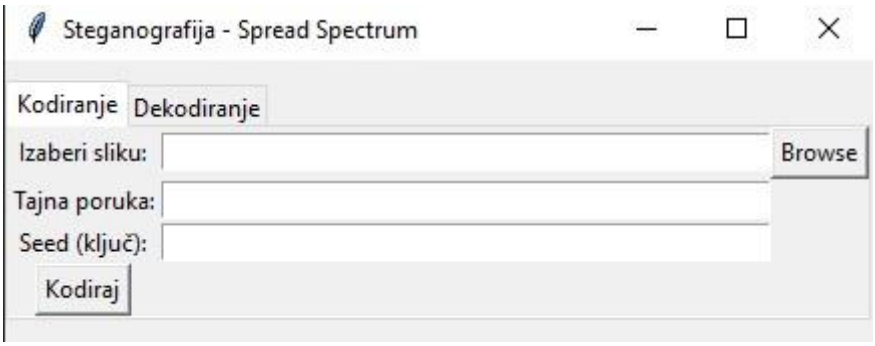
У `encode_message()` се позива функција `encode()` која кодира тајну поруку у слику. Овде се, поред путање и семена, узима и **тајна порука** из уносног поља `entry_secret_message.get()`.

У `decode_message()` се позива функција `decode()` која декодира поруку из слике. Овде се не узима тајна порука, већ се само користи путања до слике и семе за декодирање поруке.

```
103  # GUI
104  root = tk.Tk()
105  root.title("Steganografija - Spread Spectrum")
106
107  # Kreiranje tabova
108  notebook = ttk.Notebook(root)
109  notebook.pack(pady=10, expand=True)
110
111  # Tab za kodiranje
112  encode_frame = ttk.Frame(notebook)
113  notebook.add(encode_frame, text="Kodiranje")
114
115  tk.Label(encode_frame, text="Izaberi sliku:").grid(row=0, column=0)
116  entry_image_path_encode = tk.Entry(encode_frame, width=50)
117  entry_image_path_encode.grid(row=0, column=1)
118  tk.Button(encode_frame, text="Browse", command=lambda: select_image(entry_image_path_encode)).grid(row=0, column=2)
119
120  tk.Label(encode_frame, text="Tajna poruka:").grid(row=1, column=0)
121  entry_secret_message = tk.Entry(encode_frame, width=50)
122  entry_secret_message.grid(row=1, column=1)
123
124  tk.Label(encode_frame, text="Seed (ključ):").grid(row=2, column=0)
125  entry_seed_encode = tk.Entry(encode_frame, width=50)
126  entry_seed_encode.grid(row=2, column=1)
127
128  tk.Button(encode_frame, text="Kodiraj", command=encode_message).grid(row=3, column=0)
129
130  # Tab za dekodiranje
131  decode_frame = ttk.Frame(notebook)
132  notebook.add(decode_frame, text="Dekodiranje")
133
134  tk.Label(decode_frame, text="Izaberi sliku:").grid(row=0, column=0)
135  entry_image_path_decode = tk.Entry(decode_frame, width=50)
136  entry_image_path_decode.grid(row=0, column=1)
137  tk.Button(decode_frame, text="Browse", command=lambda: select_image(entry_image_path_decode)).grid(row=0, column=2)
138
139  tk.Label(decode_frame, text="Seed (ključ):").grid(row=1, column=0)
140  entry_seed_decode = tk.Entry(decode_frame, width=50)
141  entry_seed_decode.grid(row=1, column=1)
142
143  tk.Button(decode_frame, text="Dekodiraj", command=decode_message).grid(row=2, column=0)
144
145  root.mainloop()
```

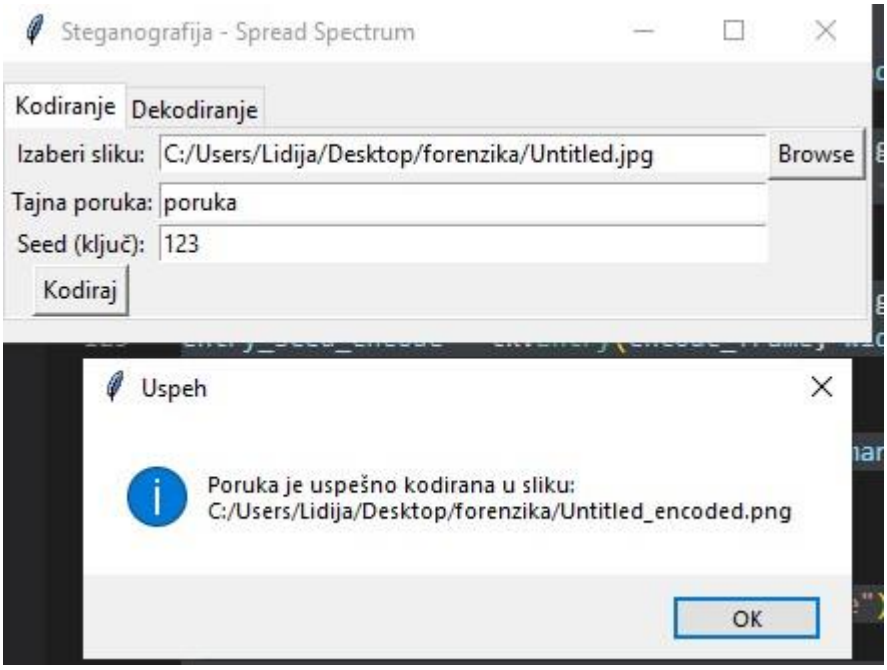
слика 7 – GUI апликација

Овај код је GUI апликација која користи *Tkinter* библиотеку за креирање графичког интерфејса за стеганографију. Апликација има два таба: један за кодирање поруке у слику и други за декодирање поруке из слике.

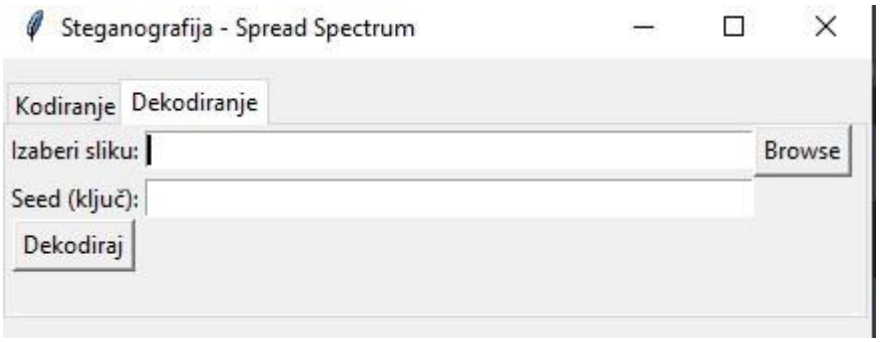


слика 8 – изглед апликације

Као што се види на слици 8, корисничи интерфејс садржи два таба: кодирање и декодирање. На слици је приказан таб за кодирање где је могуће унети слику из рачунара, затим тајну поруку као и кључ. Када се унесе све, притиском на дугме кодирај креира се фајл који садржи скривену поруку:

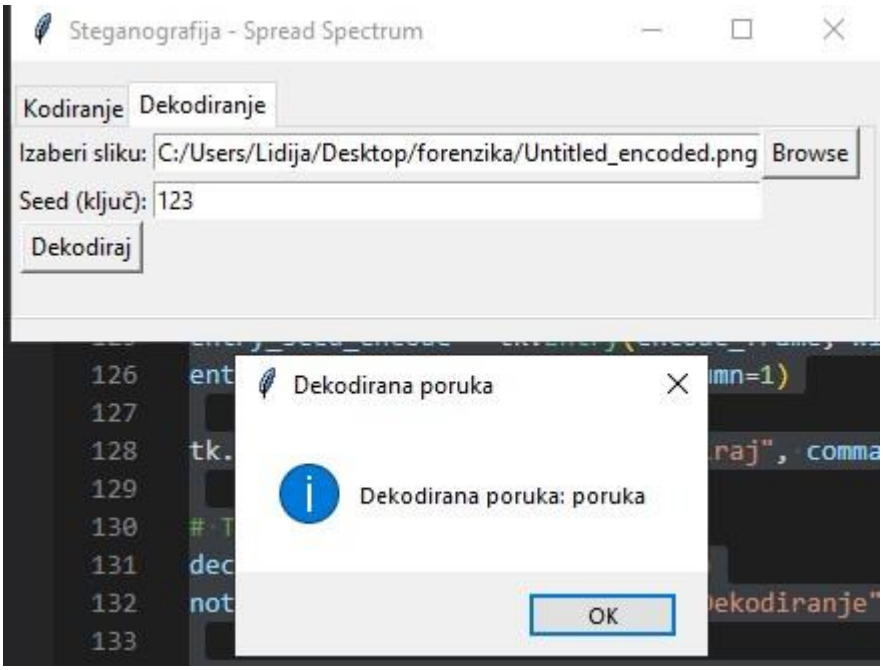


слика 9 – обавештење о успешном кодирању



слика 10 – таб за декодирање

На слици 10 приказан је иглед аплиакције када се отвори таб за декодирање. На самом табу могуће је унести кодирану слику и кључ и кликом на дугме декодирај приказује се порука на екрану:



слика 11 – обавештење о успешно декодираној поруци

слике 12 и 13 – оригинална слика и модификована слика након процеса стеганографије spread spectrum техником

Као што се види на сликама 12 и 13, ова техника не утиче на саму слику, која има готово неприметне разлике у односу на оригиналну слику.

Закључак

У овом пројекту, успешно је примењена стеганографија користећи *spread spectrum* технику као начин за сакривање тајних порука унутар дигиталних слика. Ова метода се ослања на генерисање псеудо-насумичних позиција за уметање и читање порука, чиме се повећава сигурност и отпорност на нападе. Кроз коришћење семена (*seed*), порука је скривена на начин који је изузетно тешко открити без познавања тачног кључа.

Циљ пројекта је био да се демонстрира како техника раширеног спектра омогућава прикривање информација у слици на начин који је неприметан за људско око и истовремено осигурава да је тешко разоткрити поруку без правих параметара. Процес је подељен на две главне фазе — кодирање поруке у слику и њено декодирање. У софтверском решењу коришћен је једноставан, интуитиван графички кориснички интерфејс (GUI) који кориснику омогућава лако бирање слике, унос тајне поруке и семена за кодирање или декодирање поруке.

Предност ове методе је у томе што она ефикасно користи ширину спектра дигиталних слика, што поруку чини мање осетљивом на детекцију и поремећаје. Пројекат демонстрира како применом криптографских принципа и паметног управљања ресурсима можемо остварити поуздан и сигуран систем за скривену комуникацију.

Литература

https://apps.dtic.mil/sti/tr/pdf/ADA349102.pdf - очитано 25.9.2024.

https://en.wikipedia.org/wiki/Steganography - очитано 25.9.2024.

https://www.kaspersky.com/resource-center/definitions/what-is-steganography - очитано 24.9.2024.

https://www.geeksforgeeks.org/image-based-steganography-using-python/ - очитано 26.9.2024.

https://www.techtarget.com/searchsecurity/definition/steganography - очитано 26.9.2024.

https://www.freecodecamp.org/news/what-is-steganography-hide-data-inside-data/ - очитано 26.9.2024.