UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA

DE

MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA INFORMATICĂ

Lucrare de licență

NotInSight – Aplicație de Steganografie

Absolvent

Nicolae Vlad Cornoiu

Coordonator științific

Conf. Dr. Ruxandra Olimid

București

Iulie 2020

Abstract

În prezent, securitatea și confidențialitatea datelor personale sau sensibile rămâne constant una dintre necesitățile prioritare în transferul informației pe internet. Din acest motiv, subiectul este într-o continuă dezvoltare, atât prin cercetările realizate de specialiștii în domeniu, cât și prin cele ale utilizatorilor canalelor de comunicare.

Scopul principal al acestei lucrări este acela de a furniza cunoștinte de bază cu privire la steganografia digitală dar și aprofundări cu privire la acest subiect. De asemenea, lucrarea propune o soluție pentru ascunderea informațiilor personale în transferurile de date pe internet. Mai specific, proiectul este constituit dintr-un serviciu pregătit să incorporeze date într-un fișier imagine. Acest proces este realizat printr-o serie de algoritmi aplicați asupra informației reprezentată în diferite forme: domeniul spațial și domeniul frecvențelor. Modulul conține de asemenea și soluția de decodificare a datelor secrete introduse, transformându-l într-un produs complet utilizabil pentru comunicarea pe internet.

Implementarea practicilor de steganografie a fost realizată în trei etape: transformarea datelor vizuale din domeniul spațial în cel al frecvențelor, depistarea zonelor potrivite pentru inserarea mesajului, introducerea mesajului în zonele identificate anterior și scrierea datelor sub forma compresată. Decodificarea mesajului secret este realizată prin reversul practicilor descrise anterior.

Pentru o mai mare siguranța a datelor personale ce sunt transmise mesajului privat i-a fost aplicat un algoritm de criptare.

Lucrarea prezintă in final o serie de teste și comparatii ce au fost realizate în timpul perioadei de explorare, verificând încă o dată faptul că metodele selectate sunt un bun punct de plecare în steganografia digitală.

Abstract

In the present, security and privacy stand consistently as top priorities in personal use data-transfer all around the internet. For this reason, the subject holds an ever-improving state, not only by the research undertook by domain specialists, but also by daily-basis covering techniques trials of communication channels' mere users.

The main purpose of this paper is to provide some basic knowledge of the practice of digital steganography and further insights within the aspects of this concept. Furthermore, this paper proposes a quick-to-handle solution for covering personal information throughout internet data transfer. More specific, this project introduces a microservice ready to embed a private data into an image file.

This process is materialized through a series of algorithms and procedures applied to data, both in spatial and frequency domain. This microservice is also ready to decode the secret message as well, making it feasible for the communication process on the internet.

The implementation of the steganography practices was achieved by bringing together the transformation from spatial to frequency domain, retrieval of areas which best fit the secret message, insertion of the secret message file in the aforementioned zones and writing the data in a compressed manor. Message retrieval was achieved by reversing the procedures described above.

The secret text was encrypted followed by the implementation of a shuffle algorithm process so that one may be more confident about the privacy of their transferred personal data.

The paper also presents a series of tests and comparisons that were made during the research period, proving once more that the selected methods holds a good starting point for future research in digital steganography.

Cuprins

1. Introducere

1.1. Motivație

Problema ce se situează la baza acestei lucrări este aceea a confidențialității datelor. Mai exact, este vorba despre dorința progresivă a persoanelor de a fi cât mai siguri în legătură cu păstrarea sau transferul informațiilor personale. Auzim deseori nevoia persoanelor din jur de a deține o metoda cât mai avansată de a-și păstra parolele, de a identifica un document sau o suită de date personale, de a ascunde informația față de receptori nedoriți sau de a comunica fără vreo grijă pe internet.

Desigur, pentru aceste situații, existența metodelor standardizate de criptare a datelor și informațiilor, precum și a protocoalelor și mecanismelor de comunicare ce nu permit terților să intervină poate construi un compromis destul de mic pentru majoritatea persoanelor. De asemenea, există posibilitatea nevoii de a transmite un mesaj a cărui însăși existență să fie păstrată secretă pe un canal de comunicare public. Pentru acestea, cercetarea și dezvoltarea unor noi metode de a spori siguranța utilizatorilor cu privire la datele transmise și utilizate sunt în continuă desfășurare.

1.2. Obiective

Această lucrare are ca scop prezentarea unei metode de integrare a unor informații personale într-o structură de date ce poate face obiectul unei expuneri publice. Pentru a aduce un plus metodelor precizate anterior, prezenta lucrare propune prezentarea unei aplicații construite în jurul conceptului de steganografie și utilizarea acesteia pentru a ascunde datele fără a se pune problema unor eventuale suspiciuni.

Obiectivul este ca un mesaj confidențial să fie înglobat într-o imagine inofensivă într-un mod imperceptibil ochiul uman, printr-o serie de algoritmi de compresie și manipulare a datelor.

1.3. Istoric

Exista un număr relativ restrâns de aplicații ce și-au propus dezvoltarea unor algoritmi de steganografie pentru a ascunde date.

Un prim exemplu este aplicația Xiao Steganography dezvoltată de echipa Nakasoft in Venezuela [1]. Ajunsă până la versiunea 2.6.1 în Noiembrie 2007 (ultimul release), Xiao Steganography este un utilitar cross-platform ce ii pune la dispoziție utilizatorului opțiunea de a alege metoda de criptare a datelor (ex. DES, Triple DES, MD5, SHA-1) înainte ca acestea să fie ascunse în fișierul inofensiv. Aplicația suportă doar formaturile de imagini digitale bmp și wav.

Un alt exemplu este aplicația OutGuess, dezvoltată în anul 1999 de către Niels Provos și o echipă germană [2]. În această aplicație, mesajul secret este preluat aleatoriu și introdus în zonele vizuale cele mai puțin perceptibile de către ochiul uman. De asemenea, este folosit și un principiu numit „deniable encryption" ce presupune imposibilitatea atacatorilor de a demonstra că există o informație reală în spatele unui text criptat și deci, a criptării în sine [3].

De menționat este și Steghide, aplicație ce își propune incorporarea unor date într-unul din formaturile jpeg, bmp, wav [4]. Acest util ajuns la versiunea 0.5.1. in 2003 dezvoltă un algoritm de interschimbare a valorilor pixelilor, astfel încât extracția unor anumite secvențe de informație (biți) din acești pixeli să fie cât mai apropiată de informația ce se vrea a fi introdusă (criptată cu algoritmul Rijndael – AES). Cu alte cuvinte, propune o abordare de tip „pattern-matching" pe structuri de date arborescente. Această abordare a fost aleasă pentru a rezista unor atacuri de tip statistic.

Se poate concluziona, astfel, că există un număr scăzut de aplicații ce au ca scop ascunderea datelor unui utilizator. Este observabil că acestea nu se rezumă doar la introducerea informațiilor de tip text în imagini digitale, ci oferă un spectru mult mai larg al formaturilor acceptate. Astfel, această lucrare nu are ca scop competiția cu celelalte proiecte menționate mai sus, ci găsirea și utilizarea unor metode și proceduri cu scopul introducerii datelor într-un mod cât mai imperceptibil.

1.4. Contribuție personală

Aplicabilitatea conceptelor enunțate în această lucrare au fost demonstrate printr-un serviciu pus la dispoziția utilizatorului, acesta parcurgând toți pașii de transformare și integrare a mesajului ce se dorește a fi ascuns.

O mare provocare a acestei lucrări a fost implementarea structurii pentru a construi o imagine digitala într-un format ce acceptă algoritmul de compresie JPEG. Întrucât experiența

relevantă pentru acest algoritm a fost una minimă, a fost nevoie de multă documentație pentru a înțelege modul în care acesta funcționează și cum poate fi manipulat.

Deși implementarea tuturor algoritmilor este realizată cu succes de către aplicație, nivelul de robustețe nu este la nivel de producție, astfel încât rămâne pentru moment la stadiul de dovadă a conceptelor teoretice descrise în lucrarea propriu-zisă.

1.5. Structura lucrarii

Lucrarea este structurată în 5 capitole, primul capitol fiind constituit de introducerea curentă.

Capitolul 2 are ca scop prezentarea teoriei ce a stat la baza implementarii aplicației către cititorii acestei lucrări. Pornind de la aspecte teoretice de bază despre structura steganografiei și principiile procesării imaginilor digitale, capitolul continuă cu prezentarea unor prelucrări mai avansate ale imaginii în domeniul frecvențelor urmând ca ulterior să fie discutați algoritmi de steganografie și criptografie.

Capitolul 3 precizează detaliile de implementare ale algoritmului de steganografie și descrierea aplicației practice. În subcapitolele acestuia se vor regăsi detalii despre tehnologiile auxiliare utilizate în implementarea aplicației cât și o descriere a interfeței și modului de întrebuințare al acesteia.

Capitolul 4 își propune prezentarea unor metrici de performanță standardizate pentru măsurarea zgomotului introdus într-o imagine și compararea performanțelor algoritmului prezentat în lucrarea curentă cu cele ale altor algoritmi ce incorporeaza procesul de steganografie.

Lucrarea se încheie cu Capitolul 5 ce prezintă concluzii cu privire la soluția propusă, utilitatea acesteia cât și metode de îmbunătățire pentru viitor.

2. Fundamente teoretice

2.1. Imagine digitală și procesare de imagini

Definiția unei imagini poate fi formalizată prin prisma unei funcții bidimensionale de forma $f(x,y)$ unde x și y sunt coordonate spațiale, iar valoarea funcției f pentru parametri (x,y) este reprezentată de intensitatea imaginii la coordonatele propriu-zise. Discretizând valorile x, y și $f(x,y)$ prin procesele de eșantionare (restrângerea domeniului de definiție) și cuantizare (restrângerea codomeniului), imaginea devine una digitală, constituită dintr-o serie finită de elemente cu poziție bine stabilită ce poartă numele de pixeli [5].

Procesarea imaginilor digitale (Digital Image Processing - DIP) se definește printr-un set de algoritmi ce presupun utilizarea unui calculator pentru a modifica aspectele grafice ale unei imagini. Aceste modificări sunt realizate pentru a îmbunătăți calitatea interpretării umane asupra imaginii sau pentru o un coeficient superior în percepția sistemelor computerizate [6]. DIP este o sub-categorie a procesării semnalului digital (Digital Signal Processing - DSP), mai precis o restrângere a acestuia la spațiul bidimensional.

Spre deosebire de domeniul spațial unde atât analiza cât și modificările sunt realizate direct la nivelul pixelilor în funcție de valorile lor în imaginea propriu-zisă, în domeniul de frecvență sunt reprezentate valori ce privesc rata de schimbare a valorilor pixelilor în domeniul spațial. Deși prezintă un nivel de abstractizare crescut, domeniul de frecvență oferă posibilitatea de a optimiza mult mai facil o imagine din punctul de vedere al calității, spațiului de memorie ocupat, etc. folosindu-se de conceptele procesării semnalului digital.

În practică există multe metode pentru procesarea datelor unei imagini digitale atât în domeniul spațial cât și în cel al frecvențelor. Printre aceste practici se numără Karhunen-Loeve Transform (KLT) [7], Discrete Fourier Transform (DFT) [8], Walsh-Hadamard Transform(WHT) [9] și Discrete Cosine Transform (DCT) [10].

De interes pentru această lucrare este discretizarea transformatei cosinus, DCT, datorită următoarelor avantaje:

- prezintă o capacitate crescută pentru "compactarea energiei", avantaj descries în detaliu în articolul "Discrete Cosine Transform", publicat în Ianuarie 1974 de către Nasir Ahmed, T. Natarahan și K.R. Rao [10]
- aproximează asimptotic eficiența compresiei realizată de transformata Karhunen-Loeve (ineficientă din punct de vedere computațional) [11]
- spre deosebire de DFT, coeficienții rezultați sunt decorelați, îmbunătățind atât perfomanțele computaționale cât și pe cele ale modelărilor matematice la nivel de probabilități [12].

2.2. Domeniu al frecvenței (DCT)

DCT este reprezentată de o funcție liniară, inversabilă $f : R^N \to R^N$ ce are ca scop transformarea informațiilor din domeniul spațial în cel al frecvențelor printr-o sumă de sinusoide de diferite amplitudini și frecvente [13].

DCT-II a fost propus inițial pentru compresia imaginilor digitale [14] și se definește prin ecuațiile prezentate în Formula **2.1** [15]:

$$D(i,j) = C_1(i)C_2(j)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} p(x,y)cos\left[\frac{(2x+1)i\pi}{2M}\right]cos\left[\frac{(2y+1)j\pi}{2N}\right] \qquad (2.1)$$

$$0 \le i \le M-1, 0 \le j \le N-1$$

$$C_1(i) = \begin{cases} \frac{1}{\sqrt{M}}, dacă\ i = 0 \\ \sqrt{\frac{2}{M}}, dacă\ 1 \le i \le M-1 \end{cases}$$

$$C_2(j) = \begin{cases} \frac{1}{\sqrt{N}}, dacă\ j = 0 \\ \sqrt{\frac{2}{N}}, dacă\ 1 \le j \le N-1 \end{cases}$$

unde:

- i, j – variabile discrete pentru domeniul spațial
- p(x, y) este valoarea pixelului la coordonatele x și y
- N reprezintă lungimea blocului pe care se realizează DCT
- M reprezintă lățimea blocului pe care se realizează DCT

Implicit, inversa transformatei, IDCT, este reprezentată prin ecuațiile din Formula **2.2** [15]:

$$p(x,y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} C_1(i)\, C_2(j) D(i,j) cos\left[\frac{(2x+1)i\pi}{2M}\right] cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (2.2)$$

$$0 \le x \le M-1, 0 \le y \le N-1$$

$$C_1(i) = \begin{cases} \frac{1}{\sqrt{M}}, dacă\ i = 0 \\ \sqrt{\frac{2}{M}}, dacă\ 1 \le i \le M-1 \end{cases} \qquad C_2 = \begin{cases} \frac{1}{\sqrt{N}}, dacă\ j = 0 \\ \sqrt{\frac{2}{N}}, dacă\ 1 \le j \le N-1 \end{cases}$$

Intuitiv, transformarea se traduce printr-o translație a intensității pixelilor din domeniul spațial (valorea unui pixel) către amplitudine în spațiul frecvențelor. Datorită acestei transformări putem elimina frecvențele înalte, acestea fiind greu perceptibile de către ochiul uman.

Cu alte cuvinte, reprezentativ pentru cazul prezent de analiza asupra procesării imaginilor digitale, ne vom folosi de DCT definit pe o structură de 8 x 8. Blocul de pixeli de 8 x 8 va putea fi definit ca o combinație liniară a funcțiilor cosinus ce se definesc pe aceste dimensiuni. **Figura 2.1** constituie o reprezentare grafică a acestor funcții:

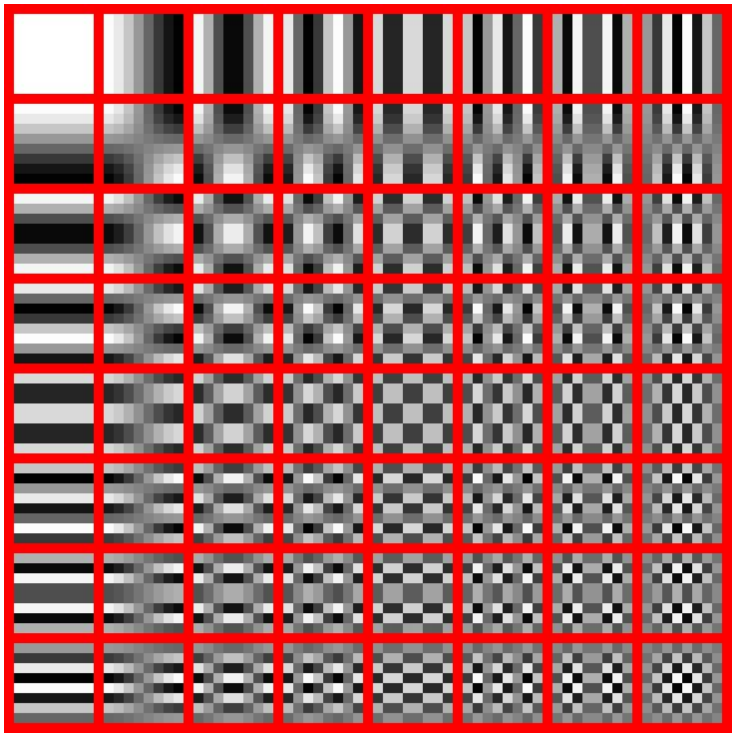

Figura 2.1 - Reprezentare grafică a funcțiilor cosinus din DCT pentru o structură de 8x8 – [16]

2.3. Steganografie

Steganografia este o metodologie de securitate cibernetică definită prin capacitatea de a incorpora informație privată într-o structură ce conține date publice. Ținta steganografiei este aceea de a transmite într-un mod cât mai sigur date cu caracter personal sau confidențiale astfel încât însăși existenta lor să nu fie cunoscută decât de către emițător și receptor.

Etimologia cuvântului "steganografie" datează încă din secolul al XVI-lea și provine dintr-o alăturare a cuvintelor grecești "steganos" (ascuns, tăinuit) și "graphe" (scriitura) [17].

În era modernă, unul din obiectivele steganografiei este acela de a trimite informație personală (fisiere text, imagini, fisiere audio, etc.) prin intermediul transferurilor de date astfel încât șansele de a detecta prezența acesteia de către metodele de steganaliza existente să tindă către 0. Alte obiective sunt constituite de creșterea robusteții datelor trimise (rezistența la diferite operații de modificare a fișierului: rescalare, compresare) și majorarea capacității informației pe care acesta o poate îngloba păstrând proprietățile mai sus descrise [18].

Un exemplu actual de aplicare a steganografiei este conceptul de "digital watermarking" (procedeu concentrat în principal pe robustețea datelor transmise).[19] Noțiunea presupune introducerea unui marcator specific într-o modalitate ascunsă și imperceptibilă, marcator ce poate atesta autenticitatea informațiilor sau chiar identitatea proprietarului informației.

O structură ce conține atât parametrii necesari cât și procedeul de bază al steganografiei se regăsește în **Figura 2.2**:

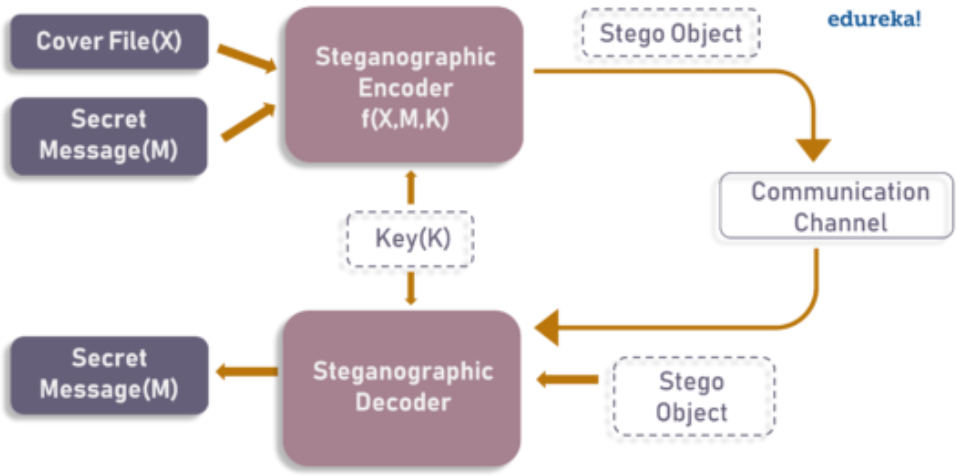

Figura 2.2 - Structura de baza a Steganografiei – [20]

Observăm că, procedural, Steganografia digitală are nevoie de o structură de date ce conține date publice, un mesaj secret, un algoritm de codificare a mesajului privat în structura de date cât și inversul acestui algoritm pentru a permite recuperarea mesajului secret de către receptor.

Una din structurile de date în care este posibilă inserarea mesajului secret este fișierul imagine, în acest caz vorbind despre steganografie în imagini digitale.

2.4. Algoritmul LSB

În ceea ce privește reprezentarea în domeniul spațial al informațiilor unei imagini digitale, cel mai nesemnificativ bit (Least Significant Bit - LSB), așa cum îi spune și numele, este bitul ce cauzează cea mai mică schimbare a datelor la nivel perceptual. Cu cât magnitudinea datelor crește, cu atât importanța sa scade. Spre exemplu, în prelucrarea unei imagini digitale reprezentată în format Red-Green-Blue (RGB), modificarea LSB-ului intensității culorii roșu dintr-un singur pixel va produce diferența prezentată în **Figura 2.2**:

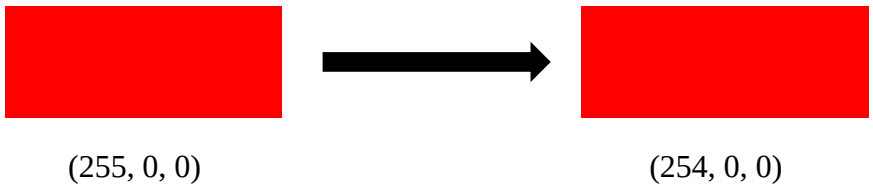

Figura 2.3 - Diferența perceptuală RGB după 1 LSB modificat (adaptata dupa [21])

Mai mult, în **Figura 2.3** se poate observa diferența atunci când în reprezentarea unui pixel în RGB sunt modificați ultimii biți pentru fiecare intensitate de culoare a acestuia:

Figura 2.4- Diferența perceptuală RGB după 3 LSB modificați

Algoritmul LSB este printre cei mai simpli algoritmi de steganografie, atât ca și concept, cât și din punct de vedere al implementării. Acesta se folosește de avantajul reprezentării informației digitale în unul formaturile cunoscute (RGB, YcbCr, CMYK, etc.) și presupune manipularea biților cel mai puțin semnificativi. Ținta acestor modificări este înglobarea informației secrete astfel încât pierderile efective ale calității imaginii să nu afecteze percepția potențialelor persoane ce intervin în procesul de comunicare și să nu creeze suspiciuni.

Algoritmul continuă până la integrarea tuturor caracterelor în imaginea inițială. Asupra imaginii din **Figura 2.5** am implementat și aplicat un algoritm LSB naiv. A fost integrat un mesaj răspândit în toți LSB ai coeficienților de culoare a pixelilor. Rezultatul se află în **Figura 2.5**:

Figura 2.5 - Female (NTSC test image) - [22]

Figura 2.6 – Fișier ce conține informație secretă

```
The cover image size is: 196748 bytes
24576 bytes of secret data were added to stego file
```

Figura 2.7 - Rezultat inserare date secrete

Se poate observa că din punct de vedere perceptual, imaginile sunt identice. Plecând de la idea că în fiecare pixel al imaginii inițiale se pot ascunde 3 biți din mesajul secret, într-o imagine de rezoluție 256x256 (aproximativ 66000 pixeli ~ 200kb) se pot ascunde 24 kb de date (echivalentul a 12 pagini dintr-o carte) fără ca acestea să fie observate de către ochiul uman. În plus, aproximativ jumătate din informația secretă nu a produs modificări la nivelul datelor din **Figura 2.5** întrucât LSB ai pixelilor coincideau cu biții ce se doreau a fi introduși.

Cu toate acestea, algoritmul naiv nu rezistă atacurilor statistice (analiza zgomotului introdus) și nici celor structurale (compresie, robustețe).

Pentru a-i îmbunătăți performanțele, algoritmului LSB i-au fost adăugate următoarele modificări:

- Criptarea mesajului secret pentru un plus de siguranță și un coeficient statistic superior
- Dezvoltarea unei metode de inserare bazată pe o parcurgere matriceală în spirală
- Introducerea algoritmului în procesul de compresie JPEG astfel încât modificările nu sunt realizate direct în domeniul spațial ci în cel al frecvențelor unde modificările nu mai sunt la fel de ușor de identificat.

2.5. Algoritmul de compresie JPEG

Prin definiție, compresia unei imagini este reprezentată de executarea unei suite de algoritmi cu scopul de a reduce costul pentru stocarea sau transferul de date. Acești algoritmi se pot folosi de avantaje ce țin de percepția vizuală cât și de proprietăți statistice pentru a-și optimiza rezultatele.

Compresiile se impart în două mari categorii [23]:

- Compresii ireversibile cu pierderea unor date considerate redundante pentru percepția umană (“lossy”), utilizate frecvent datorită compromisului realizat între calitatea imaginii și spațiul de memorie utilizat de imaginea compresată.
- Compresii reversibile fără pierdere de date (“lossless”) folosite în principal pentru situațiile inflexibile (ex. Imagistica medicală)

După cum spune și numele, tehnicile de compresie de tip “lossy” a datelor se concentrează asupra economisirii spațiului ocupat de date ulterior și nu pe fidelitatea și acuratețea datelor în procent de 100%. În mod ideal, datele pierdute sunt minime sau imperceptibile de către om.

Spre deosebire de algoritmii de compresie “lossless” a căror țintă este dezvoltarea unui proces reversibil pentru reprezentarea datelor prin exploatarea redundanței statistice, algoritmii de compresie “lossy” sunt constituiți pe teoria “rate-distortion” [24]. Aceasta teorie se bazează pe determinarea numărului minim de biți dintr-un mesaj (input) ce pot fi transferați printr-un canal

de comunicare, astfel încât datele să poată fi reconstruite la destinație fără a depăși o anumită valoare de distorsionare.

În ceea ce privește categoria de tip “lossy”, printre cei mai cunoscuți algoritmi de compresie a imaginilor digitale se numără [25]:

- Algoritmi de compresie fractală (algoritm cu o rată de utilizare scăzută din cauza performanțelor mediocre, folosit pentru comprimarea texturilor)
- Algoritmul de compresie JPEG (algoritm folosit la majoritatea imaginilor vizualizate online, find un compromis optimizat între spațiul de memorie ocupat și calitatea imaginii redate) bazat pe Discrete Cosine Transform (DCT)

De interes pentru această lucrare este algoritmul de compresie JPEG ce se materializează prin secvențializarea următorilor pași:

1. **Transformarea reprezentării culorilor din format Red-Green-Blue în format YC_BC_R (Y – componenta de luminescență, C_B – componenta de crominanță în albastru, C_R – componenta de crominanță în roșu)**

Acest pas se realizează, de obicei, printr-o înmulțire de matrici 3x3. O scurtă reprezentare grafică a transformării se regăsește în **Figura 2.8**:

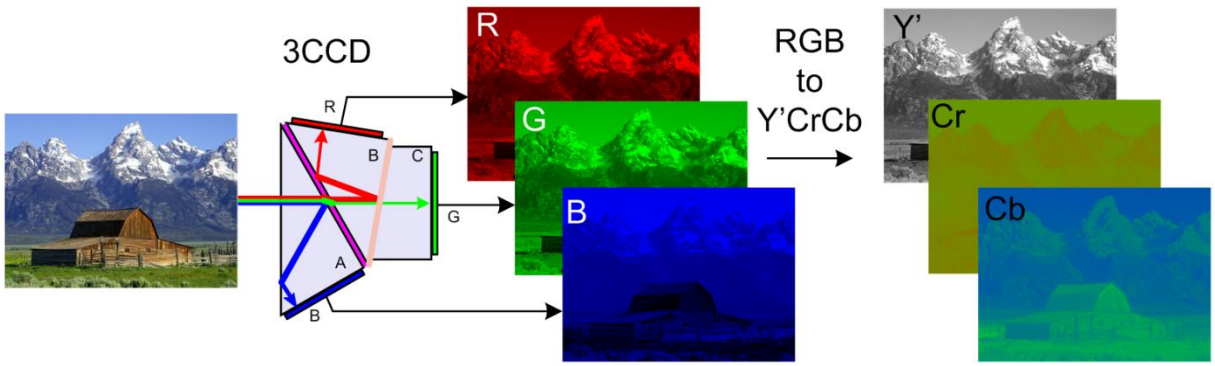

Figura 2.8 - Transformare RGB -> Y’CBCR, [26]

Algoritmul de calcul al valorilor Y, C_b si C_r este următorul [27]:

$$Y = 0.299R + 0.587\text{G} + 0.114\text{B} \quad (2.3)$$

$$C_b = 128 - 0.1687R - 0.3313\text{G} + 0.5\text{B} \quad (2.4)$$

$$C_r = 128 + 0.5R - 0.4187\text{G} - 0.0813\text{B} \quad (2.5)$$

2. Reducerea valorilor componentelor de crominanță printr-un factor de 2

Procesul poartă numele de eșantionare a crominanței (chroma subsampling) și se bazează pe sensitivitatea mai mare a ochiului uman pentru luminozitate și nu pentru culoarea informației. Cea mai întâlnită rată de eșationare în compresia JPEG este 4:2:0 și arată ca în **Figura 2.9**:

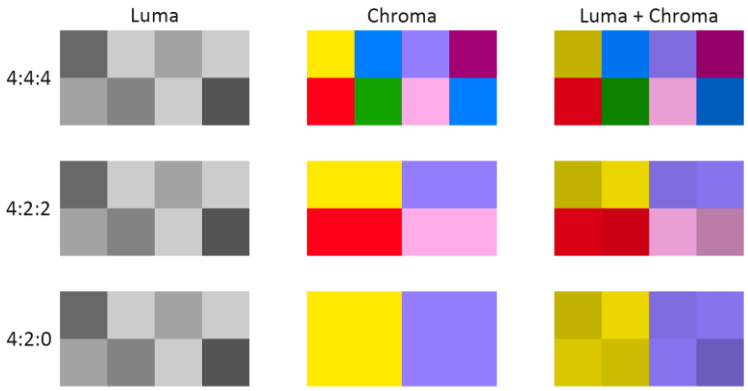

Figura 2.9 – Chroma Subsample Example: 4:4:4 – 4:2:2 – 4:2:0 – Sursa [28]

3. Aplicarea DCT pe bloc-uri adiacente de pixeli 8x8

Valorile matricilor 8x8 se încadrează in intervalul [0,255]. Urmând a fi folosită funcția cosinus ce este centrată în 0 (i.e., codomeniul este [-1,1]) vom centra în 0 și valorile matricii printr-o scădere a valorii 128.

Plecând de la formula descrisă în Subcapitolul **1.3.** aceasta devine [29]:

$$D(i,j) = \frac{1}{4}C(i)C(j)\sum_{x=0}^{7}\sum_{y=0}^{7}p(x,y)cos\left[\frac{(2x+1)i\pi}{16}\right]cos\left[\frac{(2y+1)j\pi}{16}\right] \quad (2.6)$$

$$0 \leq i,j \leq 7$$

$$C(z) = \begin{cases} \frac{1}{\sqrt{2}}, z = 0 \\ 1, \quad z = 1,2..7 \end{cases}$$

Pentru a optimiza, termenii cosinus pot fi precalculați și stocați în tabele [30].

Mai este de precizat terminologia pentru coeficienții rezultați în urma transformării. D[0][0] este numit Direct Current Term (DC) și este coeficientul de frecvență zero, obținut ca o medie a tuturor culorilor pixelilor din blocul de 8x8 transformat [31]. Restul coeficienților poartă numele de Alternating Current Terms (AC) și reprezintă componente cu frecvența crescătoare spre colțul din dreapta jos al matricei (D[8][8]).

4. Cuantizarea amplitudinilor furnizate de DCT

Componentele cu o frecvență înaltă sunt înlăturate complet, urmând a păstra doar micile variații în culoare și luminozitate pe suprafețe mai mari. Pentru aceasta, coeficienții întorși de DST sunt divizați prin valoarea corespondentă în matricea de cuantizare Q și apoi rotunjiți la întregi. Formal și particularizat pentru cazul nostru [30]:

$$Q(i,j) = Round\left(\frac{D(i,j)}{Q_{mat}(i,j)}\right) 0 \leq i,j \leq 7 \qquad (2.7)$$

unde:

- Q – matricea de valori cuantificate a coeficienților blocului după DCT
- D – matricea de valori a coeficienților bloc-ului după DCT
- Q_{mat} – matrice de cuantizare cu valori specifice în funcție de procentul de detaliu al imaginii necesar după compresie

5. Codarea rezultatelor

Aceasta este realizată printr-o metodă de parcurgere a matricii în Zig-Zag și o restrângere a datelor în funcție de lungimea șirului de numere consecutive care le preced (Rule Length Encoding). Mai precis, pentru matricea din **Figura 2.10**:

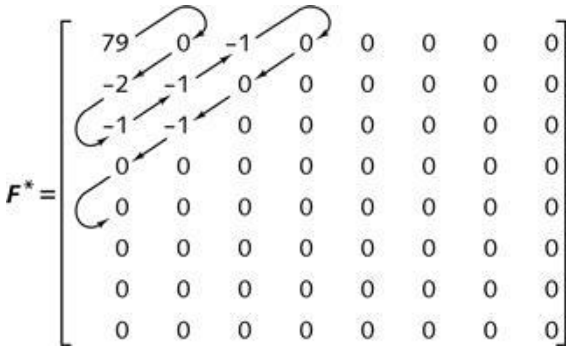

Figura 2.10 - Tehnica de alăturare a termenilor cuantizați - Sursa [32]

codarea va arăta astfel:

[(0,7), 79], [(1,2), -2], [(0,1),-1], [(0,1),-1], [(0,1),-1], [(2,1),-1], [(0,0)]

unde structurile de tipul [(a,b),c] se definesc astfel:

- a – numărul de 0-uri necontorizați aflați inaintea numărului curent
- b – numărul de biți necesari pentru a reprezenta în memorie numărul curent
- c – numărul curent

Această codare este trecută printr-o compresie Huffman (compresie de tip “lossless”) ce oferă un plus de compactare pentru economisirea spațiului pe disc [33].

Decodarea ce se realizează asupra unei imagini digitale codate cu ajutorul algoritmului de compresie JPEG este gestionată printr-o identificare a detaliilor pentru etapele de compresie și cuantizare a datelor inițiale (detalii ce se regăsesc în metadatele fișierului imagine) și parcurgerea în ordine inversă a etapelor descrise mai sus.

Performanța metodei de compresie JPEG este măsurată de eroarea dintre imaginea originală și imaginea rezultată în urma aplicării algoritmului. Statistic, metode precum MSE (Mean Square Error) sau PSNR (Peak-Signal Noise Ration) nu sunt corelate cu performanța subiectivă. Cuantizarea introduce partea semnificativă a erorii de reconstrucție, celelalte erori fiind introduse de rotunjiri ale coeficienților DCT în perioada de codare și decodare. De aceea, în practică, matricea de cuantizare este aleasă în funcție de proprietăți ale percepției umane.

2.6. Sistemul de criptare AES

Rijndael este o familie de algoritmi de criptare iterațională de tip cifru bloc (cipher block) cu o lungime variabilă a dimensiunii blocurilor și cheii de criptare [34]. AES este o specificație ce standardizează algoritmul de criptare Rijndael pentru următorii parametrii:

- Lungimea blocului de 128 biți
- Lungimea cheii de criptare: 128, 192 sau 256 biți

Selectând lungimea cheii de criptare la 256 biți (securitatea cea mai mare a acestui standard), algoritmul preia un input de 128 biți (16 bytes) și îi trece prin 14 runde de transformări succesive. Fiecare rundă este constituită din 4 operații ce sunt aplicate sub forma unor operații pe matrice [35]:

- Add Round Key (o cheie specifică fiecărei runde este construită din cheia secretă inițială, fiecare depinzând de cea creată anterior) : între input-ul rundei și cheie se realizează o operație de XOR
- Sub-Bytes : transformarea fiecărui byte pe baza unui tabel de substituție
- Shift-Rows: interschimbare valori pe fiecare rând din matrice
- Mix Columns: fiecare coloană este înmulțită cu matricea $\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$

Algoritmul Rijndael a fost ales pentru deveni AES întrucât chiar după câteva runde, conținutul inputului este complet modificat și difuzat pe întreaga structură. De asemenea, computațional, acesta nu necesită foarte multe resurse pentru operațiile de criptare și decriptare.

În ceea ce privește steganografia și studiul curent, acest algoritm a fost ales pentru a cripta datele secrete datorită securității pe care o oferă informației (utilizat în prezent pentru datele secrete ale NSA – Agenția de Securitate Națională a Statelor Unite ale Americii) cât și datorită pachetelor standard ale limbajelor de programare ce oferă o implementare a acestuia (*javax.crypto*)

3. Implementarea aplicației

3.1. Tehnologii folosite și motivația alegerii lor

Aplicabilitatea integrării unei metode de steganografie în procesul de compresie a datelor cât și posibilitatea de a recupera aceste date a fost demonstrată printr-o aplicație web ce urmează a fi detaliată în acest capitol.

Pentru a crea o aplicație web care să susțină interacțiunea cu un utilizator prin intermediul unui browser, a fost aleasă extensia Spring Boot a framework-ului de bază Spring [36]. Pentru dezvoltatorii software, Spring Boot este folosit pentru a pune la dispoziție un mediu prin care aceștia pot să creeze un microserviciu de sine stătător. Câteva din avantajele pe care acest framework le prezintă și au fost decisive în selectarea lui pentru dezvoltarea aplicației web sunt ușoara gestionare a procesării cererilor între utilizator și servicii prin intermediul sistemului de adnotări pe care acesta le definește, încorporarea unui server în dependințele framework-ului astfel încat deploy-ul este realizat automat, minimele configurări necesare pentru a pregăti aplicația și integrarea acestuia cu limbajul de programare Java.

Deși Spring este pregătit pentru a fi integrat cu o suită de limbaje de programare precum Java, Kotlin, Scala, Groovy, pentru acest proiect a fost ales limbajul Java fără prea multe dificultăți. Comunitatea developerilor este una dintre cele mai dezvoltate, iar documentația disponibilă este mult mai detaliată decât pentru celelalte. Acest limbaj oferă o utilizare facilă a diferitelor servicii gestionate cu ajutorul tool-ului Maven (tool de automatizare a descărcării și mobilizării resurselor software exterioare celor standard) întrucât și acestea au fost scrise tot în Java. Printre acestea se numără:

- *jersey-media-json-jackson*, pentru transformarea datelor din obiecte POJO în Java în format JSON, acesta fiind, alături de XML, cel mai utilizat format de date în comunicarea pe web
- *ejml*, ce definește structuri de tipul *SimpleMatrix* ce ușurează operațiile de adunare, scădere, înmulțire, inversare, etc. pe matrici. Acest pachet a fost folosit în principal pentru teste

De asemenea, în uneltele standard Java au fost regăsite următoarele pachete ce au oferit un plus în decizia de a lucra cu Java.

- *javax.crypto*, utilizat pentru operațiile de criptare și decriptare cu ajutorul *AES*
- *java.io* și *java.nio* ce au facilitat operațiile cu stream-uri de date în jurul fișierelor

Pentru interfața utilizatorului a fost folosit framework-ul open-source Vaadin. Acesta oferă capacitatea de a implementa interfețe grafice direct în limbajul Java, iar integrarea cu Spring Boot este una facila. Întrucat atenția acestui proiect s-a îndreptat mai mult spre realizarea algoritmului de steganografie, Vaadin a fost o alegere ușoara pentru a crea rapid un mediu ușor accesibil și prietenos pentru utilizator.

VBinDiff este un alt produs software ce a fost utilizat pe parcursul dezvoltării acestei aplicații [37]. Acesta este un tool open-source ajuns la ultima versiune in 2017, folosit pentru compararea în forma hexazecimală a datelor fișierelor de până la dimensiuni de 4GB. Acesta a fost vital în nevoia de a examina datele secrete introduse până în momentul în care a fost dezvoltat și algoritmul de decodificare.

Un alt utilitar software folosit pentru creșterea productivității a fost Postman [38]. Cu ajutorul acestuia am reușit să creez request-uri de tip REST către serviciile de procesare a imaginilor fără a fi nevoie de a porni de fiecare dată interfața grafică.

În final, pentru partea de măsurare a performanțelor algoritmului propus în această lucrare s-a experimentat cu limbajul de programare Python [39] și librării ale acestuia precum *OpenCV* [40] (pentru analizarea imaginilor) și *matplotlib* [41] (pentru reprezentarea graficelor).

3.2. Modul de utilizare

În această secțiune va fi prezentat modul de utilizare al aplicației. Interfața grafică realizată este una simplistă, dar îmbogățită de o serie de verificări astfel încat utilizatorului nu i se oferă ocazia de a periclita buna funcționare a algoritmilor din spate.

Primul ecran al aplicației, prezentat în **Figura 3.1** este creat pentru a-i oferi utilizatorului posibilitatea de a decide dacă își dorește să înceapă procesul de codare sau pe cel de decodificare.

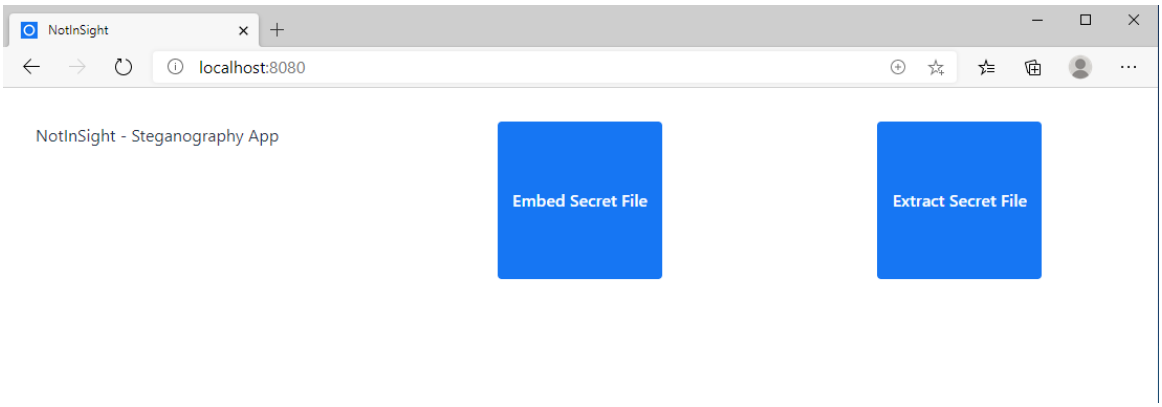

Figura 1.1 - ecran inițial aplicație NotInSight

Alegând opțiunea “Embed Secret File”, utilizatorul se găsește pe o pagină unde își poate încărca un fișier ce conține datele secrete cât și pe cel în care vor fi integrate. Acesta are posibilitea de a încărca fișierele prin intermediul butoanelor “Upload Cover File” și “Upload Secret File” sau mai simplu, prin *drag & drop* în căsuța specifică, după cum se poate observa în **Figura 3.2**:

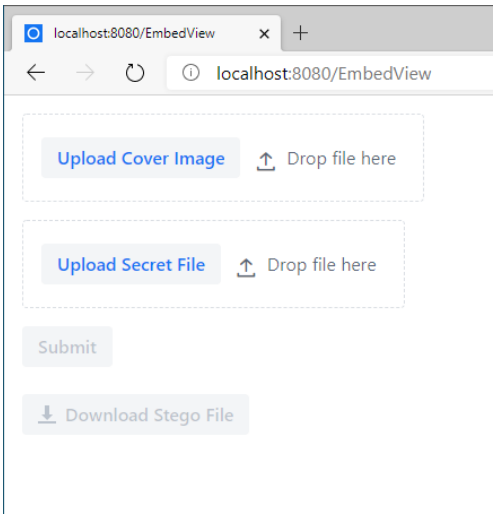

Figura 3.2 - Ecran Embed aplicație NotInSight

Atât în partea de frontend cât și în partea de backend există o serie de verificări ce nu vor permite utilizitorului să încarce un format de date neacceptat de serviciile de procesare a imaginii. De asemenea, pentru a păstra o fidelitate cât mai mare a fișierului stego în raport cu imaginea inițială, utilizatorului nu îi este permis a încărca un fișier secret a cărui dimensiune să depășească ~ 10-15% din dimensiunea fișierului cover. Se poate observa în **Figura 3.3** o notificare ce va avertiza utilizatorul despre nevoia de a schimba datele de intrare:

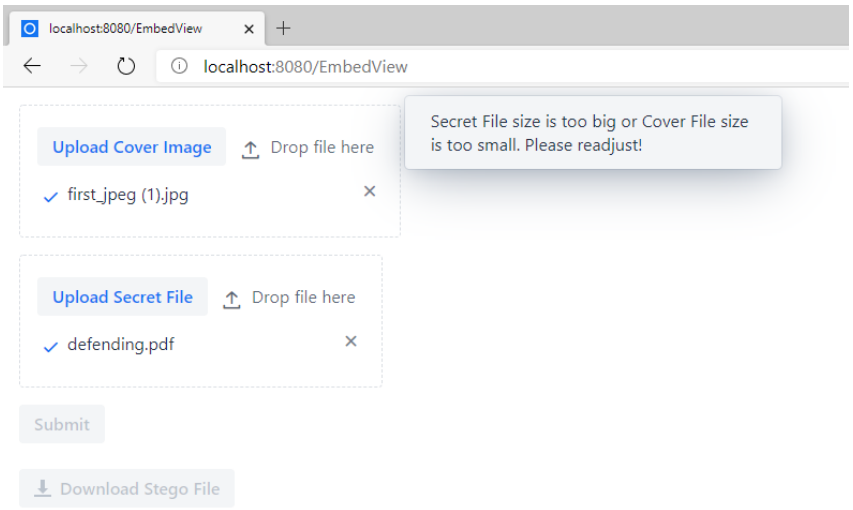

Figura 3.3 - Avertisment dimensiune fișiere

După ce au fost încărcate cele două fișiere, butonul de "Submit" devine disponibil, iar utilizatorul poate să trimită datele pentru a fi procesate. În momentul în care fișierul stego a fost creat, este afișată o notificare (**Figura 3.4**) , butonul de descărcare devine disponibil, iar utilizatorul poate descărca fișierul.

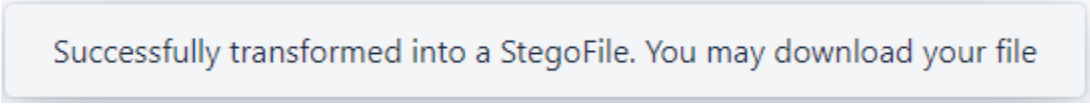

Figura 3.4 - Notificare codare reușită

În ecranul "Extract", același gen de ecran îl întâmpină pe utilizator și îi oferă posibilitatea de a încărca un fișier ce conține date secrete. După ce acestea au fost extrase, utilizatorul este notificat de terminarea procesului și are posibilitatea de a descărca fișierul, după cum se poate observa în Figura **3.5**:

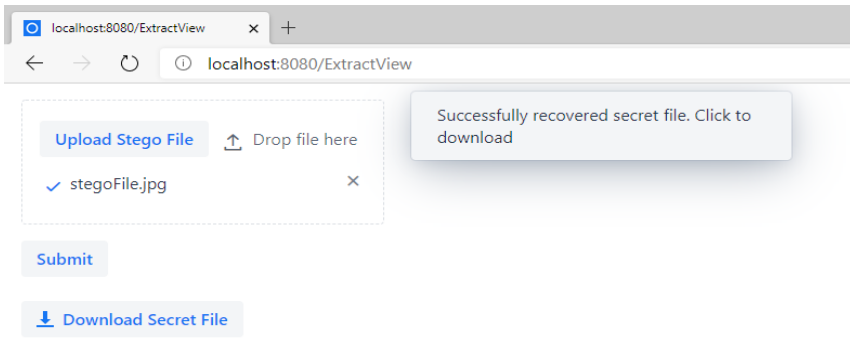

Figura 3.5 - Notificare recuperare cu succes a mesajului secret

3.3. Detalii de implementare

Aplicația este publică și disponibila la link-ul de GitHub [42].

Pentru început, informația grafică a fișierului imagine a fost încărcată folosind librăria ImageIO (*javax.imageio.ImageIO*). Cu ajutorul acesteia, aplicația acceptă majoritatea formatelor de fișier imagine, aceasta fiind transformată la final în format JFIF (JPEG File Interchange Format), format ce acceptă algoritmul de compresie JPEG.

Informația fișierului secret a fost stocată ca șir de bytes. Asupra acestui șir de bytes a fost aplicat algoritmul de criptare AES, utilizând librăria *javax.crypto* din pachetul *java.base*. Dupa câteva experimente cu librăria, am folosit modul Cypher-Block Chaining (CBC). Un generator de numere aleatoare (Secure Random) a fost folosit pentru a crea vectorul de inițializare necesar criptării primului bloc de 16 bytes de text în clar. Textul criptat rezultat în urma aplicării AES este folosit ca vector de inițializare pentru cel de-al doilea bloc de criptat, al doilea text criptat este folosit pentru ce de-al treilea bloc ș.a.m.d.. Deși neparalelizabilă la criptare (timpul de execuție crește), din punct de vedere al securității datelor, această metodă este mult mai sigură decât altele precum Electronic Codebook (ECB) ce generează un output identic pentru un același input [43].

După criptarea mesajului secret începe procesarea imaginii. Acesteia îi este aplicată o transformare din spațiul de culori RGB în YCbCr folosind formulele **2.3**, **2.4**, **2.5**, urmată apoi de procesul de downsample. Acesta a fost realizat pentru componentele Cb și Cr printr-o reducere succesivă a coeficienților pe verticală și orizontală, rezultatul fiind media aritmetică a acestora.

Etapa de scriere a segmentelor și markerilor specifici unei imagini JPEG a fost una anevoioasă. Segmentele de date conțin informații necesare decompresării imaginii (tabelele de cuantizare a luminescenței și crominanței, tabelele Huffman, etc.). Întrucât majoritatea detaliilor despre informațiile conținute de aceste segmente se găsesc în hexazecimal, am folosit programul VBinDiff pentru a putea identifica rapid informațiile scrise într-un fișier in limbaj hexazecimal. Site-ul [44] a fost foarte util pentru a putea testa coerența acestor date din header-ul imaginii.

Pentru DCT, inițial am implemetat formula clasică ce a dus la complexitate $O(n^4)$ per bloc de 8x8. Imaginile de calitate înaltă ajung la dimensiuni foarte mari: la o imagine de 1920x1080 vorbim despre 32400 blocuri de 8x8 numai pentru una din componentele Y, Cb sau Cr -> aproximativ 132 milioane de operații per componentă doar pentru transformarea DCT. A trebuit

astfel căutată o metodă alternativă pentru a diminua această complexitate. Am găsit, astfel, formula AAN propusă de Arai, Agui și Nakajama [45]. Aceasta se bazează pe proprietatea de periodicitate a funcției cosinus și pe posibilitatea de a precalcula termenii ce folosesc această funcție. Urmărind informațiile disponibile pe site-ul [46] (grup ce gestionează librăria open-source pentru compresia JPEG) am reușit să verific că și aceștia folosesc o versiune a acestui algoritm.

Mesajul secret a fost inserat folosind o tehnică mai puțin convențională pentru dispersarea acestuia în întreg pachetul de coeficienți DCT rezultați în urma transformării. Primele informații introduse au fost cele ce precizează lungimea fișierului secret, dimensiunea (în bytes) a extensiei acestuia și extensia fișierului în sine. Coeficienții au fost așezați în serii de matrice pătratice astfel încât așezarea lor să nu fie una liniara și predictibilă. Pentru a reduce distorsiunea produsă imaginii au fost evitați coeficienții de tip DC cât și cei egali cu 0.

Compresia Huffman a fost realizată cu ajutorul unor tabele standard. Înainte de a descoperi existența markerilor și segmentelor de date aferente în structura unei imagini digitale JPEG am încercat implementarea unor arbori Huffman. Deși implementarea a fost realizată cu succes, nu am reușit să integrez și o metodă prin care să transpun și codificarea în detaliile imaginii, astfel încat sistemul de operare să le poata citi și astfel să decompreseze fișierul. Aici, de ajutor a fost aplicația JPEGSnoop ce oferă informații detaliate și foarte bine structurate cu privire la tabelele de compresie, cuantizare ș.a. [47]. Am preluat astfel o serie de imagini digitale într-un format ce folosește compresie JPEG atât din surse proprii cât și din arhive de pe internet pentru a compara tabelele pe care acestea le folosesc. Am ales astfel o serie de tabele standard, folosite de cele mai multe camere digitale și produse software pentru editare de imagini.

În urma acestor procedee, fișierul Stego este creat și livrat utilizatorului. Acesta are posibilitatea de a-l descărca și trimite pe un mediu de comunicare. Imaginea digitală este in format JPEG File Interchange Format(JFIF), unul dintre cele mai folosite de pe internet [48].

Ajunsă la destinatar, acesta poate încărca imaginea în aplicație pentru a fi trimisă către serviciile ce rezolvă decodarea mesajului secret, aplicând pașii descriși anterior în ordine inversă: citirea markerelor și segmentelor de date ce conțin informația necesară codării (tabele cuantizare, tabele compresie, detalii despre dimensiuni, etc.), decodarea Huffman pentru a obține coeficienții DCT și extragerea mesajului criptat din aceștia. Decriptarea este realizată cu ajutorul AES, iar mesajul secret este recuperat.

4. Experimente și performanțe

4.1. Metrici de performanță

Metricile de performanță ce privesc imaginile digitale reprezintă un subiect destul de controversat în analiza calității unui fișier imagine întrucât foarte multe dintre acestea nu sunt considerate ca fiind corelate cu subiectivismul. Procesul de determinare a acurateței unei imagini se numește IQA (Image Quality Assessment) și presupune atât o analiză obiectivă cât și una subiectivă.

Peak Signal Noise Reduction (PSNR) este una dintre cele mai cunoscute metode de aproximare a distorsiei, fiind un raport între puterea maximă a unui semnal și puterea zgomotului ce perturbă reprezentarea acestuia [49].

Structural Similarity (SSIM) este un index ce măsoară factorul de similitudine între două imagini bazându-se pe schimbări în informația perceptuală [50].

Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) este o metrică antrenată pe imagini similare cu cea care se dorește a fi măsurată și identifică corelarea între spațiu și obiecte [51].

Pentru a testa fidelitatea fișierelor Stego, am selectat 3 imagini ce fac obiectul studiilor și analizelor asupra imaginilor digitale și am comparat rezultatele aplicării procesului de inserare a mesajului secret cu cele ale altor algoritmi ce oferă posibilitatea de a integra informație secretă într-un fișier imagine.

Pentru a ne asigura că toți algoritmii operează la capacitate maximă a fost selectat un text de aprox. 3KB pentru teste.

Imaginile de referință sunt următoarele:

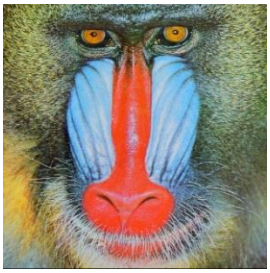

Figura 4.1 - Baboon - [52]

Figura 4.2 - Airplane (F-16) - [53]

Figura 4.3 - Peppers - [54]

În comparație cu alți algoritmi ce realizează steganografie, rezultatele obținute pentru aceste imagini au fost prezentate în Tabelul **4.1**:

SecretFile = 3KB		MSE	PSNR	SSIM	BRISQUE
Xiao Steganography	Baboon	0.047	66.168	1.000	18.600
	Peppers	0.047	66.163	1.000	18.501
	Airplane	0.047	66.204	1.000	8.149
HideNSend	Baboon	52.321	36.264	0.986	16.490
	Peppers	17.425	40.619	0.978	10.299
	Airplane	19.519	40.105	0.982	20.515
NotInSight	Baboon	48.875	36.470	0.985	13.111
	Peppers	21.138	39.817	0.976	12.104
	Airplane	18.646	40.284	0.985	22.667
LSB	Baboon	95.647	34.630	0.982	12.328
	Peppers	95.462	35.231	0.928	18.202
	Airplane	102.396	34.945	0.960	24.194

Tabel 4.1 - Comparare algoritmi steganografie

Este bine de ținut cont de faptul că imaginile prelucrate cu majoritatea produselor software de steganografie nu sunt rezistente la tipul de compresie JPEG atunci când sunt transmise pe internet, acesta constituind un factor important în ceea ce privește fiabilitatea produsului rezultat.

În Figurile **4.4** – **4.9** se pot observa histogramele realizate asupra imaginilor inițiale și asupra celor prelucrate cu algoritmul NotInSight. Acestea demonstrează o deviație minimă în ceea ce privește procentul statistic realizat asupra coeficienților pixelilor.

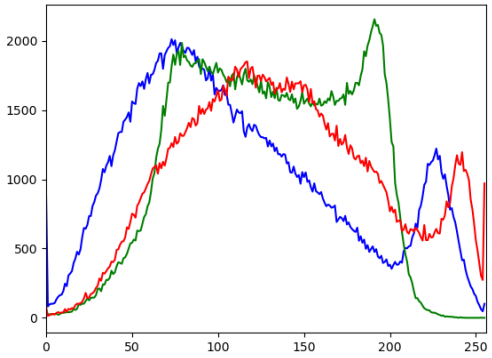

Figura 4.4 - Histograma Baboon

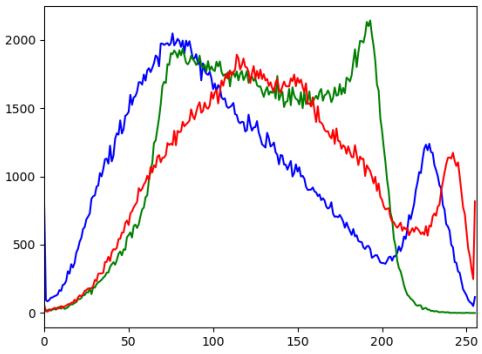

Figura 4.5 - Histograma Stego Baboon

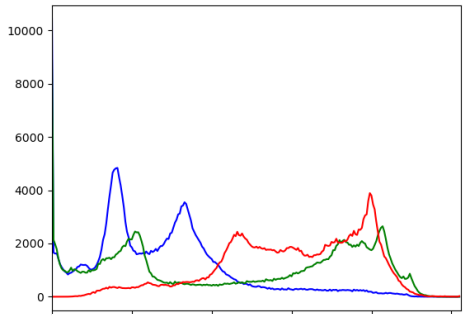

Figura 2.6 - Histograma Peppers

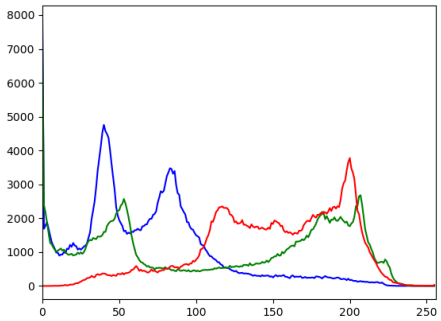

Figura 4.7 - Histograma Stego Peppers

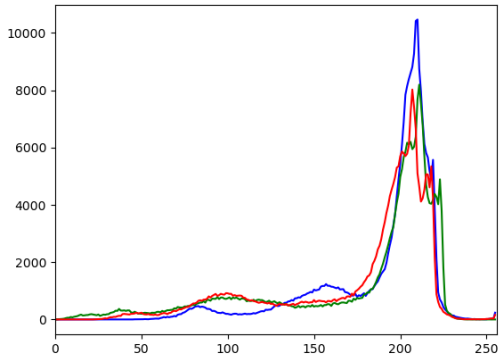

Figura 4.8 - Histograma Airplane

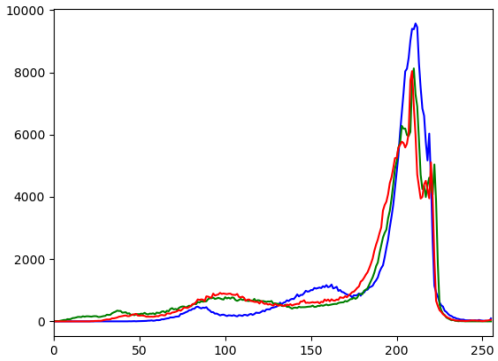

Figura 4.9 - Histograma Stego Airplane

5. Concluzii și direcții viitoare de dezvoltare

Steganografia digitală este un domeniu amplu al ascunderii informațiilor pe internet. Acesta este într-o continuă dezvoltare, fiind limitat doar de creativitatea persoanelor ce implementează aceste practici, după cum a fost verificat și în lucrarea curentă.

Am dezvoltat, astfel, un produs ce este capabil de a extrage informația grafică dintr-o imagine într-un format oarecare urmată de preluarea unui fișier cu date confidențiale de la utilizator. Informația grafică a fost trecută printr-un proces de transformare din domeniul spațial în cel al frecvențelor, iar datele confidențiale au fost criptate și ascunse în coeficienții obținuți sub acest nivel. În final, a fost aplicat un algoritm de compresie ce corespunde standardului JPEG astfel încât imaginea rezultată este pregatită de a fi transferată pe orice mediu de comunicare. Decodificarea mesajului secret este posibilă prin executarea pașilor în ordine inversă, implementați de asemenea în cadrul aplicației.

Deși execuția algoritmilor descriși în această lucrare ating scopurile propuse în debutul lucrării, aplicația prezintă și o serie de îmbunătățiri ce fac obiectul unei dezvoltări ulterioare.

Deși arhitectura actuala a aplicației implică execuția frontend-ului și a backend-ului pe același sistem de gestionare a datelor, fișierele încărcate de utilizator (în special cel secret) ar putea fi criptate înainte de a se realiza transferul către partea de servicii. Acest lucru ofera securitate suplimentara în cazul unui atac de tip Eavesdropping [55]. În același sens, o conexiune securizată cu *https* ar putea împiedica atacuri mai avansate (Man in the Middle [56]).

De asemenea, asupra interfeței grafice se pot aduce o serie îmbunătățiri ce ar putea crește calitatea experienței utilizatorului atunci când folosește aplicația.

În concluzie, lucrarea constituie un punct de plecare în direcția modalităților de ascundere a informației într-o imagine digitală, oferind explicația detaliată a pașilor parcurși pentru realizarea acesteia.

Bibliografie

[1] List of freeware, 35 Best Free Steganography Software For Windows, Disponibil online: https://listoffreeware.com/list-of-best-free-steganography-software-for-windows/, Data accesării 27.06.2020

[2] Niels Provos, OutGuess, Disponibil online: https://web.archive.org/web/20150831083519/http://outguess.org/, Data accesării 27.06.2020

[3] Canetti, R., Dwork, C., Naor, M., & Ostrovsky, R., Deniable encryption. In Annual International Cryptology Conference (pp. 90-104). Springer, Berlin, Heidelberg. (1997, August).

[4] Sourceforge, Steghide - manual, Disponibil online: http://steghide.sourceforge.net/documentation/manpage.php , Data accesarii: 02.07.2020

[5] Gonzalez, Rafael C. "Richard E. Woods Digital Image Processing, Pearson." (2018).

[6] McAndrew, Alasdair. *A computational introduction to digital image processing*. Chapman and Hall/CRC, 2015.

[7] R.D.Dony "Karhunen-Loeve Transform" The Transform and Data Compression Handbook, Ed. K. R. Rao and P.C. Yip, Boca Raton, CRC Press LLC, 2001

[8] Rao, Kamisetty Ramamohan, Do Nyeon Kim, and Jae Jeong Hwang. *Fast Fourier transform-algorithms and applications*. Springer Science & Business Media, 2011.

[9] Wikipedia, the free encyclopedia, Hadamard transform, Disponibil online: https://en.wikipedia.org/wiki/Hadamard_transform, Data accesării: 02.07.2020

[10] Ahmed, Nasir, T_ Natarajan, and Kamisetty R. Rao. "Discrete cosine transform." *IEEE transactions on Computers* 100.1 (1974): 90-93.

[11] Cranley, Nicola. Handbook of research on wireless multimedia: quality of service and solutions. Ed. Liam Murphy. Information Science Reference, 2009.

[12] Nuno Vasconcelos, Discrete Cosine Transform, UCSD, Disponibil online: http://www.svcl.ucsd.edu/courses/ece161c/handouts/DCT.pdf, Data accesării: 02.07.2020

[13] Rao, K. Ramamohan, and Ping Yip. Discrete cosine transform: algorithms, advantages, applications. Academic press, 2014.

[14] Ahmed, Nasir. "How I came up with the discrete cosine transform." Digital Signal Processing 1.1 (1991): 4-5.

[15] Stanford Edu, The Discrete Cosine Transform (DCT), Disponibil online: https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/dct.htm, Data accesării: 02.07.2020

[16] https://en.wikipedia.org/wiki/File:Dctjpeg.png, Data accesării: 29.06.2020

[17] Merriam-Webster, "Steganography.", Disponibil online: https://www.merriam-webster.com/dictionary/steganography , Data accesării: 29.06.2020

[18] Cheddad, Abbas, Joan Condell, Kevin Curran and Paul Mc Kevitt,. "Digital image steganography: Survey and analysis of current methods." Signal processing 90.3 (2010): 727-752.

[19] Cox, I., Miller, M., Bloom, J., Fridrich, J., & Kalker, T., "Digital watermarking and steganography", Morgan kaufmann, 2007.

[20] Choudary, A. *Steganography Tutorial | A Complete Guide for Beginners | Edureka.*, 2019 Accesibil online: https://www.edureka.co/blog/steganography-tutorial, Data accesării: 29.06.2020

[21] Boite Aklou, Steganography Tutorial: Least Significant Bit (LSB), Disponibil online: https://www.boiteaklou.fr/Steganography-Least-Significant-Bit.html, Data accesarii: 02.07.2020

[22] http://sipi.usc.edu/database/database.php?volume=misc&image=1#top, Data accesării: 26.06.2020

[23] Singh, Manjari, Sushil Kumar, Siddharth Singh, Manish Shrivastava "Various image compression techniques: Lossy and lossless." International Journal of Computer Applications 142.6 (2016): 23-26.

[24] Blau, Yochai, and Tomer Michaeli. "Rethinking Lossy Compression: The Rate-Distortion-Perception Tradeoff." *arXiv preprint arXiv:1901.07821* (2019).

[25] Wikipedia, Lossy Compression, Disponibil online: https://en.wikipedia.org/wiki/Lossy_compression, Data accesării: 29.06.2020

[26] https://en.wikipedia.org/wiki/YCbCr#/media/File:CCD.png, Data accesării: 29.06.2020

[27] Eric Hamilton, JPEG File Interchange Format, 1992, Disponibil online: https://www.w3.org/Graphics/JPEG/jfif3.pdf , Data accesării: 29.06.2020

[28] Jack, Keith. *Video demystified: a handbook for the digital engineer*. Elsevier, 2011

[29] William Buchanan, DCT (Discrete Cosine Transform), Disponibil online: https://asecuritysite.com/comms/dct2, Data accesării 29.06.2020

[30] Cabeen, Ken, and Peter Gent. "Image Compression and Discrete Cosine Transform", College of Redwoods, 2008.

[31] CSE 228 Week 3 Part 1, UC San Diego, Jacobs School of Engineering, Disponibil online: https://cseweb.ucsd.edu/classes/sp03/cse228/Lecture_5.html, Data accesării: 01.07.2020

[32] Poynton, Charles. *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012.

[33] Huffman, David A. "A method for the construction of minimum-redundancy codes." *Proceedings of the IRE* 40.9 (1952): 1098-1101.

[34] Daemen, Joan, and Vincent Rijmen. "AES proposal: Rijndael." (1999).

[35] FIPS, PUB. "197." Advanced encryption standard (AES) 26 (2001).

[36] Spring Team, Spring Boot, Disponibil online: https://spring.io/projects/spring-boot, Data accesării: 27.06.2020

[37] Christopher J. Madsen, Vbindiff, Disponibil online: https://www.cjmweb.net/vbindiff/, Data accesării: 27.06.2020

[38] Postman, Inc., Postman, Disponibil Online: https://www.postman.com/, Data accesării: 29.06.2020

[39] Python, Documentație oficială Python, Disponibil online: https://www.python.org/doc/, Data accesării: 02.07.2020

[40] OpenCV, Documentație oficială OpenCV, Disponibil online: https://opencv.org/, Data accesării: 02.07.2020

[41] Matplotlib, Documentație oficială Matplotlib, Disponibil online: https://matplotlib.org/contents.html, Data accesării: 02.07.2020

[42] NotInSight, Disponibil online: https://github.com/VladCornoiu/NotInSight-SteganographyApp, Data accesării: 02.07.2020

[43] Block cipher mode of operation, Wikipedia, the free encyclopedia, Disponibil online: https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation, Data accesării: 02.07.2020

[44] AE27FF, JPDump, Disponibil online: https://cyber.meme.tips/jpdump/#, Data accesării: 02.07.2020

[45] Arai, Yukihiro, Takeshi Agui, and Masayuki Nakajima. "A fast DCT-SQ scheme for images." *IEICE TRANSACTIONS (1976-1990)* 71.11 (1988): 1095-1097.

[46] Independent JPEG Group, Accesibil online: http://www.ijg.org/, Data accesării: 02.07.2020

[47] Calvin Haas, JPEGsnoop 1.8.0 - JPEG File Decoding Utility, Disponibil online: https://www.impulseadventure.com/photo/jpeg-snoop.html, Data accesării: 01.07.2020

[48] JPEG JFIF, W3, Disponibil online: https://www.w3.org/Graphics/JPEG/, Data accesării: 01.07.2020

[49] Wikipedia, Peak signal-to-noise ratio, Disponibil online: https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio, Data accesării: 28.06.2020

[50] Wikipedia, the free encyclopedia, Structural Similarity, Disponibil online: https://en.wikipedia.org/wiki/Structural_similarity Data accesării: 02.07.2020

[51] Mathworks, Brisque, Disponibil online: https://www.mathworks.com/help/images/ref/brisque.html, Data accesării: 02.07.2020

[52] http://sipi.usc.edu/database/database.php?volume=misc&image=10#top, Data accesării: 28.06.2020

[53] http://sipi.usc.edu/database/database.php?volume=misc&image=11#top, Data accesării: 28.06.2020

[54] http://sipi.usc.edu/database/database.php?volume=misc&image=13#top, Data accesării: 28.06.2020

[55] Lyna Griffi, Eavesdropping in Computer Security: Definition & Laws, Disponibil online: https://study.com/academy/lesson/eavesdropping-in-computer-security-definition-laws.html, Data accesării: 02.07.2020

[56] Wikipedia, the free encyclopedia, Man-in-the-middle attack, Disponibil online: https://en.wikipedia.org/wiki/Man-in-the-middle_attack, Data accesării 02.07.2020