Analiza sistemului software

“ASI”

(Aplicatie de steganografie in imagini)

*Realizată de: Data: 12.10.2017*

Balea Ana-Maria

An studiu: 4 IS, grupa 1.1

*Cuprins:*

1. Prezentarea cerintelor

1.1.Functionale

1.2.Non-functionale

1.3. Cerinte care influenteaza arhitectura

1. Componente

2.1.Descompunerea in component

2.2. Definirea responsabilitatilor componentelor si a relatiilor dintre ele. Argumentare

1. Prezentarea sistemului software din doua perspective
2. Indicatori de calitate

4.1.Identificarea celor mai importanti 3 indicatori de calitate

4.2. Specificarea masurii alese pentru fiecare indicator de calitate. Argumentarea alegerii

1. Identificarea tehnologiilor middleware folosite pentru a comunica intre componente. Argumentarea alegerilor
2. Identificarea pincipalelor modele si stiluri arhitecturale folosite. Argumentarea alegerilor
3. Prezentarea scenariilor de validare a arhitecturii

**Ce e, de fapt, Steganografia?**

In IT, steganografia reprezinta metoda de a ascunde mesaje (fisiere) in alte fisiere mai mari si anume in imagini de tip jpg, bmp, in fisiere audio (mp3 sau wav) sau chiar video (avi) fara a exista posibilitatea ca o terta persoana sa stie sau sa afle de existenta lor. **Steganografia** nu trebuie confundata cu **criptografia**. Acesta din urma face ca un mesaj sa devina indescifrabil, dar existenta lui este vizibila, iar steganografia ascunde existenta mesajului si nu mesajul.

Ceea ce vom analiza in continuare este metoda de steganografie in imagini. Pentru o imagine "true color" fiecare pixel este reprezentat pe 24 de biti adica cate un octet pentru fiecare nuanta de rosu, verde si albastru (RGB - Red Green Blue).  
Daca vom schimba pentru fiecare byte bitul cel mai putin semnificativ al fiecarui pixel cu un bit care compune mesajul pe care vrem sa-l ascundem, atunci efectul asupra fiecarui pixel este atat de mic incat imaginea modificata ramane practic aproape identica cu originalul. In realitate ochiul uman nu poate sesiza vreo diferenta intre imaginea modificata (in care se ascunde mesajul) si cea originala.

***Scopul urmarit: se doreste analiza sistemului software “ASI”***

1. ***Prezentarea cerintelor***
   1. *Functionale:*

* Ca urmare a faptului ca aplicatia este de tip Desktop, se impune ca sistem de operare Windows/Linux.
* Este necesara instalarea mediului de dezvoltare integrat Microsoft Visual Studio in vederea functionarii aplicatiei.
* Aplicatia nu va rula in cazul in care versiunea Visual Studio este mai veche fata de cea pe care ea a fost implementata.
* Interfata cu utilizatorul va fi prevazuta cu :
* Buton de inserare a imaginii in care va fi ascuns textul, respectiv de salvare a imaginii care sufera modificari.
* Buton de inserare a textului care urmeaza sa fie ascuns in imagine, sau dupa caz, de salvare a textului care a fost ascuns in imagine in format necriptat.
* Caseta de tip text in care va fi introdusa o parola de minim 6 caractere( fara aceasta, textul introdus nu va putea fi ascuns! )
* Butoane de ascundere a textului in imagine, respectiv de extragere a textului din imagine.
  1. *Cerinte non-functionale:*
* **Integritatea:** in urma rularii aplicatiei, informatia introdusa in imagine va fi returnata nealterata.
* **Constrangeri tehnologice**: ca urmare a faptului ca este o aplicatie de tip desktop, ea nu va putea fi rulata decat de pe PC-uri.
* **Securitate**:aplicatia trebuie sa asigure utilizatorul ca informatia ascunsa in spatele imaginii, va ramane “secreta” in fata oricaror altor utilizatori.
* **Disponibilitate:** Dacă o aplicație nu este disponibilă atunci când este nevoie de ea, atunci este puțin probabil că aplicația își îndeplinește rolul pentru care ea a fost dezvoltată. Spre exemplu, Aplicațiile Internet trebuie însă să fie disponibile 24 din 24. Disponibilitatea poate fi măsurată ca și raportul de timp în care aplicația este utilizabilă.Fiind o aplicatie de tip Desktop, ea nu ridica probleme similare celor enuntate mai sus.
* **Testabilitate:**arhitectura aplicatiei este una relativ simpla, astfel incat testarea acesteia nu ridica probleme.

* 1. *Cerinte care influenteaza arhitectura:*
* Integritatea
* Testabilitate
* Securitate

**2. Componente**

*2.1. Descompunere in componente*

* Serverul de autentificare
* Interfata cu utilizatorul
* Importa
* Exporta
* Modul criptare/decriptare
* Modul ascundere informatie
* Modul extragere informatie

*2.2.Definirea responsabilitatilor componentelor. Argumentare*

* **Serverul de autentificare**: aceasta componenta preia rolul de intrare in “sistem” si de monitorizare in vederea unei desfasurari cu succes a procesului de autentificare(login) al utilizatorului .
* **Interfata cu utilizatorul**:odata trecut de procesul de autentificare, utilizatorul are acum acces la interfata. Intr-o alta ordine de idei, putem spune ca interfata reprezinta “telecomanda” prin intermediul careia utilizatorul va putea lansa comenzile dorite, apasand butoanele puse la dispozitie: importare imagine/text/, exportare imagine/text, inserare parola pentru criptare/decriptare, criptare/decriptare text, ascundere informatie, extragere informatie.
* **Importa** : aceasta componenta are required un path catre fisierul de unde va fi preluat textul si ofera un array de stringuri (textul de ascuns).
* **Exporta** : aceasta componenta functioneaza invers fata de cea anterioara, avand required un array de stringuri(textul care tocmai a fost ascuns) si ofera un path catre fisierul de unde textul ascuns va putea fi preluat ulterior.
* **Modul criptare/decriptare**:aceasta este una dintre componentele cheie in ceea ce priveste procesul de ascundere a informatiilor in imaginea importata. Textul va fi criptat prin intermediul algoritmului AES. Din motive de securitate suplimentare, acest modul nu va putea fi realizat fara introducerea unei parole de minim 6 caractere. Odata ce este criptat, textul ascuns va fi salvat intr-un fisier a carui path va fi dat la alegere de catre utilizator, pentru a putea fi extras ulterior.
* **Modul ascundere informatie**: aceasta componenta ocupa un rol decisiv in vederea stabilirii unei functionari corecte sau nu a aplicatiei, conform principiului steganografiei. Mai exact, vom putea spune ca acest modul a fost implementat corect, daca din momentul ascunderii informatiei, aceasta nu este vizibila de catre niciun alt utilizator.
* **Modul extragere informatie**: la fel ca si componenta anterioara, ea este decisiva in a putea stabili daca aplicatia functioneaza respectand conditiile impuse de procesul de steganografie pe imagini. Astfel, in momentul in care utilizatorul solicita extragerea informatiei ascunse in imagine, aceasta trebuie sa fie intacta, fara sa isi fi modificat continutul . In caz contrar, aplicatia nu functioneaza corect.

3. ***Prezentarea sistemului software din doua perspective***

***4.Indicatori de calitate***

*4.1.Identificarea celor mai importanti 3 indicatori de calitate:*

* Disponibilitatea
* Testabilitate
* Securitate

4.2. Specificarea masurii alese pentru fiecare indicator de calitate. Argumentarea alegerii

* Disponibilitate: Aplicatia fiind de tip Desktop, Intr-adevar aplicatia este disponibila ori de cate ori un utilizator solicita accesarea ei
* Testabilitate: Analizand arhitectura aplicatiei, se poate observa ca aceasta este simpla si in acelasi timp bine structurata, cu un numar moderat de componente si implicit, nu ridica probleme la intelegerea ei. Astfel, testarea ei urmeaza un fir simplu de pasi, in asa fel incat cel care testeaza aplicatia, sa isi poata da seama cu usurinta daca ea functioneaza corect sau nu.
* Securitate: In urma unei testari complete a aplicatiei, putem observa ca informatia odata ascunsa in imagine, nu mai este vizibila de catre nimeni altcineva, cu exceptia utilizatorului curent. In momentul solicitarii extragerii informatiei ascunse, ea este returnata fara sa fi suferit modificari. Asadar, constatam ca sunt respectate principalele cerinte referitoare la securitate(autentificare, autorizare, criptare, integritate, nerepudiere). Consideram acest indicator de calitate ca fiind unul dintre cei mai importanti, deoarece este esential in aplicatiile bazate pe steganografie. Absenta lui indica o functionalitate incorecta a aplicatiei.

***5.******Identificarea tehnologiilor middleware folosite pentru a comunica intre component. Argumentarea alegerilor***

***6.******Identificarea principalelor modele si stiluri arhitecturale folosite. Argumentarea alegerilor***

*Stil architectural: Analizand arhitectura aplicatiei, aceasta tinde sa se incadreze cel mai mult spre* ***tipul bazat pe inlantuire(pipelining)*** *datorita structurii bazate pe date de intrare( informatia de ascuns) si procedeul urmat pana la obtinerea datelor de iesire( informatia ascunsa, urmata de extragerea ei).*

7. Prezentarea scenariilor de validare a arhitecturii

Intrucat aplicatia este de tip Desktop, scenariile de validare se rezuma la cazul banal de verificare, acesta constand in urmatoarele etape:

* Utilizatorul se va autentifica
* Odata autentificat, el are acces la interfata de comenzi
* Vom importa o imagine in care urmeaza sa ascundem informatia, respectiv un fisier text
* Pentru a putea cripta textul, este obligatorie introducerea unei parole de minim 6 caractere
* Dupa introducerea parolei, este permisa ascunderea informatiei in imagine. Tot aici, utilizatorul dispune de optiunea de salvare a informatiei ascunse in imagine in varianta necriptata.
* Pentru extragerea informatiei, utilizatorul este din nou obligat sa introduca aceeasi parola introdusa in momentul ascunderii informatiei, din motive de securitate
* Daca la extragerea informatiei, se constata ca aceasta este intacta, continutul ei nesuferind modificari, atunci putem spune despre aplicatie ca intruneste toate conditiile impuse si implicit, ca arhitectura sa este una valida. In caz contrar, ea nu respecta termenii de validare.