Esteganografia en contingut multimèdia

Alejandro Miñano Sicilia
ETIS

Consultor: Antoni Martínez Ballesté

Desembre 2005

INDEX DE CONTINGUTS

1. Especificació del problema

L'objectiu d'aquest Treball Fi de Carrera (TFC) és el disseny i la implementació d'un mètode per a amagar, de forma secreta, un missatge en un contingut multimèdia (imatge, àudio, vídeo, etc.). El missatge serà una tira de caràcters que caldrà emmagatzemar al contingut multimèdia de forma robusta, entenent-se per robust aquell mètode que permet recuperar el missatge tot i haver modificat "lleugerament" l'objecte.

2. Fonaments

L'esteganografia és l'art de comunicar de manera oculta un missatge. La paraula esteganografia ve de les paraules gregues *steganos* (impenetrable, reservat, discret) i *graphia* (escriptura), significant doncs literalment escriptura encoberta.

L'esteganografia ha estat utilitzada durant segles com a mitjà per a ocultar informació sensible de "mirades indiscretes" abans d'arribar a la seva destinació i es basa en què qui ha d'enviar la informació i qui l'ha de rebre acorden un mètode mitjançant el qual la informació en qüestió serà ocultada i, per tant, algun sistema de comunicació *a priori* és necessari perquè l'esteganografia sigui útil.

Esteganografia i criptografia són sovint considerades com la mateixa cosa. Tot i que totes dues poden coexistir en una mateixa transmissió d'informació, no es tracta del mateix. Tant l'esteganografia com la criptografia s'utilitzen per a protegir la informació, però mentre que la primera s'ocupa d'ocultar-la fent que aquesta informació esdevingui "invisible", la criptografia s'ocupa d'encriptar-la de manera que sigui inintel·ligible. En altres paraules, amb la criptografia, el no destinatari pot veure que s'està enviant una determinada informació, tot i que, en principi, no serà capaç d'entendre-la, però amb l'esteganografia ni tan sols, també en principi, se n'adonarà de què l'intercanvi d'informació està tenint lloc. Un dels principals avantatges de l'esteganografia és el fet que el contingut de cobertura no sembla ser res més que el que és, com per exemple una fotografia o una cançó, mentre que un fitxer encriptat està cridant "continc informació sensible!!". Si un fitxer encriptat és descobert en trànsit o emmagatzemat, indicarà que el seu contingut està intencionadament ocult i pot motivar a algú a desxifrar el seu contingut. A més a més, un fitxer encriptat pot també indicar que hi haurà més informació sensible viatjant pel mateix camí en el futur i la connexió en qüestió pot ser monitorejada per a tractar de determinar patrons en el tràfic que ajudin a desxifrar les dades. El fet de tot això és que si un hagués de monitorejar el tràfic que viatja per una determinada connexió, la informació encriptada serà molt més fàcil de veure que la informació que estigui oculta a dintre d'un fitxer no encriptat.

L'esteganografia es basa en el fet que els sentits humans són insuficients si els comparem amb l'anàlisi que pot portar a terme una màquina o fins i tot els sentits d'altres animals. El sistema visual o auditiu humans no poden detectar canvis molt subtils o minúsculs en presentacions visuals o auditives, fent així de l'esteganografia un mitjà efectiu per a l'ocultació d'informació privada o sensible. Un factor molt important en el que també es basa l'esteganografia és que una persona no té per què saber que un fitxer amb una imatge (o amb una cançó, o un bloc de text, etc.) conté informació oculta. És doncs un mitjà molt més efectiu per a protegir la informació si l'atacant (receptor no intencionat o no autoritzat) desconeix que el material que té al davant conté informació oculta perquè, en cas contrari, l'esteganografia queda oberta a l'atac i perd el seu més potent avantatge: la "invisibilitat".

2.1 Una mica d'història

La història proporciona innumerables situacions a les quals una determinada informació havia de travessar territoris enemics o hostils per a arribar a la seva destinació sense ser detectada. A través dels temps, la gent ha utilitzat enginyosos mitjans per a ocultar informació i, a mesura que passava el temps, aquests mètodes s'anaven millorant donat que els anteriors s'anaven descobrint.

Alguns d'aquests exemples són:

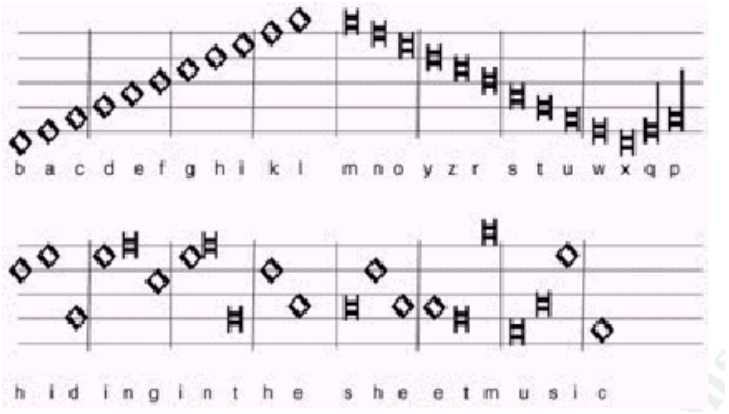
- A la Grècia antiga, s'utilitzava un mètode mitjançant el qual una persona era escollida com a missatger i se li afaitava el cap. A continuació, se li tatuava el text secret al cap i s'esperava a què li tornés a créixer el cabell. El missatger podia llavors viatjar a la seva destinació i passar qualsevol inspecció ja que aparentment no portava res sospitós. Un cop davant del destinatari, se li tornava a afaitar el cap per a poder llegir el missatge. Un dels inconvenients d'aquest mètode era el temps necessari per a fer arribar el missatge a la seva destinació, donada la necessitat d'haver d'esperar a què el cabell creixés el suficient per a ocultar el text abans de poder enviar el missatge.

- També a la Grècia antiga, s'utilitzava un mètode basat en taules cobertes de cera. Es treia la cera de la taula, s'escrivia el missatge a la fusta i es tornava a recobrir de cera. El receptor de la taula només havia de tornar a treure la cera per a poder llegir el missatge.

- També un grec, Aenas l'Estratega va idear un mètode pel qual es perforaven en un disc de fusta forats representant lletres de l'alfabet grec. A continuació s'havia de passar un fil pels forats per a fer visible el missatge.

També va utilitzar un altre mètode consistent en fer petits forats a sobre de les lletres d'un document per a formar un missatge. Aquests forats minúsculs eren generalment indetectables per qualsevol que no sabés que hi eren i que simplement estigués llegint el document. Aquest mètode encara s'utilitza i és conegut com el codi del diari.

- Un mètode força enginyós és el que Gaspar Schott detalla al seu llibre *Schola Steganographica*. Consisteix a codificar la informació assignant lletres a notes musicals. En aparença, un es trobarà davant d'una partitura normal i corrent, però si un tracta de tocar la peça amb algun instrument, el més probable és que el resultat no sigui molt agradable a l'oïda. Aquí podem veure un exemple:



- Durant la Segona Guerra Mundial, s'utilitzaven tintes invisibles per a ocultar informació en memoràndums o cartes. Exemples de tintes invisibles són la llet, el vinagre i el suc de fruites. Cadascuna d'aquestes substàncies s'obscureix amb la calor i eren especialment útils en aquella època donada la seva fàcil disponibilitat.

- Una altra forma d'enviar missatges era escrivint un text on després el destinatari havia d'agafar una determinada lletra de cada paraula per a obtenir el missatge que en realitat es volia transmetre. Per exemple, si del missatge següent:

> Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.

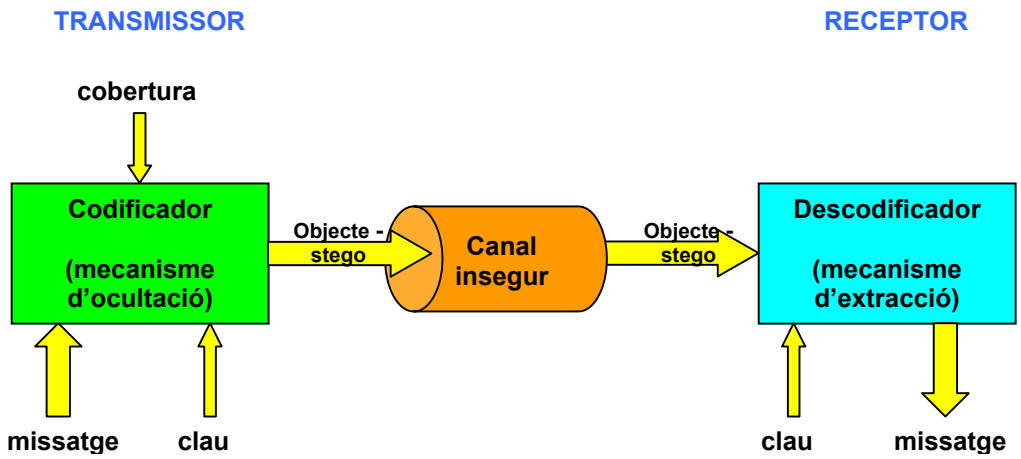
Prenem la segona lletra de cada paraula obtenim el missatge que realment es vol enviar:

> Pershing sails from NY June 1.

- També durant la Segona Guerra Mundial es va utilitzar el micropunt. Un missatge secret era fotogràficament reduït fins a la mida d'un punt i pegat com el de la lletra i en un paper contenint un missatge escrit. La transmissió del missatge secret quedava doncs oculta.

2.2 Com funciona?

L'esteganografia moderna consisteix generalment en ocultar informació a dintre de fitxers que continguin imatges o música. Aquests tipus de fitxers poden contenir informació perceptualment irrellevant o redundant, podent així ser substituïda per missatges ocults. El diagrama següent mostra, a grans trets, el funcionament bàsic de l'esteganografia digital:



La cobertura és el fitxer original amb la imatge, l'àudio o el vídeo. El missatge és allò que s'hi vol ocultar. La clau és el paràmetre que serveix per a encriptar el missatge abans de la seva ocultació i l'objecte-stego és el fitxer resultant que conté el missatge ocult. S'ha de tenir cura de quanta informació s'oculta al contingut: a major longitud del missatge, més gran és la modificació que s'ha de fer al fitxer de cobertura i per tant més probabilitat hi haurà de què les modificacions puguin ser detectades amb anàlisis estadístics.

Podem trobar tres tipus de tècniques:

- Substitució
- Injecció
- Generació de nous fitxers

2.2.1 Tècnica 1: substitució

Els fitxers poden contenir àrees de dades insignificants o sense utilitzar. Aquestes àrees poden ser doncs substituïdes sense canvis perceptibles a la claredat visual o auditiva del contingut del fitxer. Això ens permet d'ocultar informació sensible a dintre del fitxer i que el fitxer en qüestió sigui aparentment el fitxer original, sense modificació alguna. El mètode de substitució del bit menys significatiu (*Least Significant Bit* o LSB) es basa en la modificació del darrer bit d'un byte. La teoria és que, simplement reemplaçant aquest bit en cada byte, el canvi no serà perceptible per l'ull o l'oïda humanes, en funció del tipus de fitxer.

Suposem que els següents tres bytes representen el color d'un determinat pixel d'una imatge de 24 bits (el primer byte pot representar la quantitat de vermell, el segon la de verd i el tercer la quantitat de blau):

10010110 01101010 11100101

Suposem ara que canviem el primer bit (1) del primer byte. Tenint en compte que aquest bit és el més significatiu, el canvi implicarà un efecte considerable a la imatge i podria ser fàcilment detectat. Si el que canviem, en lloc de ser el primer bit, és el darrer o LSB (0), el més probable és que aquest canvi sigui imperceptible. Així doncs, a l'hora d'ocultar un missatge en un fitxer amb una imatge podríem utilitzar els LSB.

El LSB funciona millor en fitxers que tenen "molt de soroll". Per exemple, imatges amb molts colors i formes, o fitxers d'àudio que tenen molts sons diferents i efectes com ara ecos. I això pel fet de que canvis als LSB canvien els valors dels bytes i per tant els colors o sons. En conseqüència, a més soroll en un fitxer, més difícil serà per a l'ull o l'oïda humanes percebre petits canvis.

Un punt molt important és que el mètode de la substitució no varia la mida del fitxer de cobertura, sempre i quan es controli que el missatge a ocultar hi càpiga. La tècnica del LSB és molt utilitzada en aplicacions esteganogràfiques donat que és ràpida i fàcil d'utilitzar. Un important desavantatge del LSB és que és molt sensible a canvis en el fitxer.

2.2.2 Tècnica 2: injecció

La injecció implica inserir el missatge secret directament a l'objecte de cobertura. El problema d'aquesta tècnica és que generalment incrementa la mida del fitxer en qüestió. Tot i que això no ha de ser un gran problema si un atacant no té una còpia del fitxer original, no deixa de ser un desavantatge de la tècnica. La major part dels programes actuals tenen mètodes per a situar dades als fitxers en àrees que seran ignorades o no visualitzades en obrir el fitxer.

2.2.3 Tècnica 3: generació de nous fitxers

Aquesta tècnica implica agafar el missatge secret i utilitzar-lo per a generar un nou fitxer des de zero. Un avantatge important d'aquesta tècnica és que mai hi haurà un fitxer "original" amb el qual comparar el nou fitxer, fet que incrementa la probabilitat de què es no detectin les dades ocultes a dintre del contingut.

2.3 Elecció del fitxer de cobertura

Quin format digital escollir per a ocultar els missatges privats? L'esteganografia pot ser utilitzada en pràcticament tot tipus de fitxers, tot i que aquí comentarem només els mètodes utilitzats amb imatges, àudios i vídeos. Dels mètodes que es comentaran a continuació, s'ha de tenir en compte que alguns d'ells poden ser utilitzats en tots tres tipus de continguts.

2.3.1 Imatges

2.3.1.1 *Least Significant Bit* (LSB)

Ja hem avançat aquest mètode a la secció 2.2.1. Per a un ordinador, una imatge no és més que un array de números que representen intensitats de llum en diferents punts o pixels. Les imatges digitals són normalment de 24 o de 8 bits per pixel. Les de 24 bits són conegudes com a imatges de color vertader o *true color images*, i són més apropiades per a l'ocultació de missatges donats els bytes d'informació addicionals. No obstant això, donats aquests bytes addicionals, ocupen també més que les de 8 bits, amb el que la transferència a través d'Internet d'imatges de gran mida pot aixecar sospites sota determinades circumstàncies. És per això que a vegades s'escullen imatges de 8 bits com a fitxer de cobertura, preferentment en escala de grisos perquè les transicions entre colors són més suaus i per tant els canvis dels LSB són més difícilment perceptibles.

2.3.1.2 Emmascarament

La compressió d'imatges pot afectar la integritat del missatges ocults. Tenim dos tipus de compressió:

- Amb pèrdua (*lossy*): JPEG (*Joint Photographic Experts Group*) utilitza aquest sistema i ofereix el ratio de compressió més elevat.

- Sense pèrdua (*lossless*): BMP (*Microsoft Bitmap*) i GIF (*Graphics Interchange Format*) són dos formats que proporcionen una major qualitat però una menor compressió, essent per tant continguts més propicis per a ocultar-hi missatges.

Una simple conversió des d'un format GIF o BMP a un format de compressió amb pèrdua com ara el JPEG pot destruir la informació oculta a la imatge. Tècniques d'emmascarament o de filtratge són més efectives que el LSB quan s'utilitzen imatges JPEG.

Cobrint o emmascarant un senyal dèbil però perceptible amb un altre, per fer el primer imperceptible, explotem el fet que el sistema visual humà no pot detectar canvis petits en certs dominis d'una imatge. Això ens retorna al fet que la percepció humana no detecta canvis minúsculs de color. En qualsevol cas, aquest sistema d'ocultació de missatges és més proper al marcatge (*watermarking*) que a l'esteganografia, tot i que el concepte és el mateix.

La DCT (*Discrete Cosine Transform*) és un mètode que lliga bé amb l'ocultació en imatges JPEG i serà comentat a la secció 2.3.3.1.

2.3.2 Àudio

De la mateixa manera que en el cas d'imatges és important saber si la imatge es distribueix en paper i posteriorment s'escaneja o si procedeix directament d'un suport electrònic, en el cas de senyals d'àudio s'ha de tenir present quin camí ha seguit des de la seva generació fins el destinatari. El senyal pot ser transmès digitalment o analògicament, propagar-se per un cable o per l'aire, etc. Les característiques del senyal més robustes a la transmissió i, per tant més adequades per a l'ocultació de missatges, dependran de les condicions en què s'hagi transmès el senyal. Anàlogament al cas de les imatges, la quantitat d'informació que es podrà ocultar serà més gran quan major sigui la velocitat de transferència (major freqüència de mostreig i/o major número de bits per mostra). En general, la transferència d'informació oculta que es pot aconseguir està compresa entre 2 i 128 bits per segon. Alguns dels mètodes per a l'ocultació d'informació en senyals d'àudio són els següents:

- Modificació del LSB
- Codificació de fase
- Tècniques d'espectre eixamplat
- Introducció d'eco
- Emmascarament perceptual

2.3.2.1 Modificació del LSB

Consisteix en alterar el bit menys significatiu de cadascuna de les mostres de senyal. Idealment, això permet introduir 1 kbps per cada kHz de freqüència de mostreig. No obstant això, aquest sistema introdueix un soroll de fons que pot resultar perceptible, especialment a les parts del senyal amb menor amplada. En canvi, si el senyal presenta un volum o soroll inherent elevat (per exemple, en la retransmissió d'un esdeveniment esportiu), el soroll queda emmascarat pel propi senyal. Aquest mètode és simple, però poc robust a sorolls de canal i manipulacions del senyal.

2.3.2.2 Codificació de fase

El sistema auditiu humà és insensible a canvis absoluts en la fase del senyal. D'aquesta forma es pot ocultar informació en la fase dividint un senyal d'àudio en segments (trames) i substituint la fase del segment inicial per la informació a ocultar. Per a evitar la degradació del senyal, s'haurà de mantenir els canvis relatius de fase entre segments. Una possible codificació d'un missatge binari és utilitzar el següent codi: $\Phi=\Pi/2$ equival a un "0" i $\Phi=-\Pi/2$ equival a un "1". Per això es pot realitzar una descomposició del senyal mitjançant la transformada discreta de Fourier o DFT i modificar la fase absoluta de cadascuna de les components freqüencials obtingudes (a cada component freqüencial ocultem un bit d'informació). Aquest mètode requereix limitar la transferència del senyal a ocultar, donat que en cas contrari l'alteració entre les fases de les diferents components freqüencials provoca distorsió per dispersió de fase. Es poden aconseguir transferències entre 8 i 32 bps.

2.3.2.3 Tècniques d'espectre eixamplat

Es tracta d'un mètode anàleg a l'utilitzat en sistemes de comunicacions. En general, en comunicacions interessa concentrar el missatge a transmetre en una amplada de banda el més limitada possible, per a obtenir un ús eficient del canal de comunicacions. Això no obstant, transmetre la mateixa potència en una amplada de banda més elevada presenta avantatges, com ara una major robustesa contra interferències o desvaneixements del canal, així com la possibilitat de realitzar transmissions per sota del nivell de soroll, privacitat, etc. Una de les tècniques més utilitzades rep el nom de seqüència directa, que consisteix a distribuir el missatge en un ampli marge de freqüències realitzant el producte del senyal a transmetre per un senyal pseudoaleatori. El procés de recuperació de la informació requereix la clau utilitzada per a generar la seqüència aleatòria. El senyal eixamplat en banda, que té una aparença sorollosa, s'atenua a un valor aproximat del 0,5% del marge dinàmic del senyal que albergarà la informació, i se sumen tots dos senyals.

D'aquesta manera es limita l'efecte del soroll introduït. Mitjançant aquesta tècnica es pot aconseguir una transferència d'informació oculta d'aproximadament 4 bps.

2.3.2.4 Introducció d'eco

Consisteix en introduir eco en el senyal que albergarà la informació. Les dades a ocultar es codifiquen en l'amplitud inicial, decaïment i retard de l'eco. Si se suma el senyal original que albergarà la informació amb una versió retardada del mateix senyal, és possible aconseguir que l'eco afegit soni com a reverberació, el que pot no ser molest i passar inadvertit (la pròpia sala d'audició afegeix una reverberació depenent de les seves dimensions i característiques físiques) La quantitat de retard que es pot afegir sense que es percebi com a un efecte molest depèn de la qualitat de l'enregistrament, tipus de so, etc. En general, si el retard és inferior a 1 ms. no es notarà l'efecte. Per a codificar la informació oculta es poden utilitzar dos retards, un per l'1 binari i l'altre pel 0. Els seus retards seran inferiors a 1 ms. i es fixarà una amplada inicial i decaïment del senyal retardat a sumar que evitin que l'eco sigui audible. Per a codificar més d'un bit es pot dividir el senyal en trames i afegir un senyal diferent a cada trama en funció de la informació a codificar. Per a recuperar la informació oculta es poden utilitzar les eines clàssiques de detecció de periodicitats, com l'autocorrelació, el cepstrum, etc. Existeixen diverses estratègies addicionals per a mitigar la reducció de qualitat deguda a la introducció de la informació oculta, com ara la monitorització de l'amplada del senyal en funció del temps, per a controlar que l'amplada del senyal introduït sigui sempre menor a un determinat percentatge del senyal original.

2.3.2.5 Emmascarament perceptual

Ja hem esmentat que l'oïda humana no és molt sensible a sons diferencials. L'emmascarament perceptual és el concepte d'amagar un so darrera d'un altre més fort de la mateixa freqüència. Hem vist aquesta tècnica essent utilitzada molt sovint, com per exemple, quan algú que no vol ser escoltat per ningú més que el seu interlocutor legítim posa en marxa sons auxiliars com ara la televisió o un equip de música per a emmascarar la seva conversa. Doncs bé, aquesta tècnica pot ser utilitzada de la mateixa manera en esteganografia i és efectiva donades les debilitats en el sistema auditiu humà.

2.3.3 Vídeo

L'esteganografia en vídeo utilitza generalment com a mètode de manipulació la transformada discreta de cosinus. (*Discrete Cosine Transform* o DCT). Un bon exemple pot ser la vídeo conferència, on es requereix un elevat ritme de fotogrames, el que sovint implica un gran estrès sobre les xarxes digitals. Per a solucionar aquest problema, s'utilitza compressió diferencial amb pèrdua (*lossy*), el que significa que únicament les diferències en cada fotograma són transmeses.

Aquest mètode elimina el tema de la comparació d'imatges per a determinar diferències que poden portar al descobriment de la utilització de l'esteganografia.

2.3.3.1 *Discrete Cosine Transform* (DCT)

L'ocultació de missatges en imatges és considerada com una forma efectiva d'ocultar dades sensibles. Això no obstant, la compressió de les imatges destruiria la integritat del missatge ocult, deixant-lo irrecuperable. El següent mètode resumeix com fan alguns programes per a superar aquest punt.

La DCT funciona utilitzant quantització a les parts menys importants de la imatge pel que fa a les capacitats visuals humanes. Quantització significa per exemple que el valor 5,7489763 pot ser arrodonit a 6 i per tant representat per molt menys bits. Per descomptat, fer això a cada valor produiria una notable distorsió de la imatge. No obstant això, sota condicions normals, l'ull humà no detecta altes freqüències a les imatges, la qual cosa permet a la DCT fer grans modificacions en aquestes freqüències amb una distorsió poc apreciable. La DCT treballa dividint la imatge en àrees més petites i fent la quantització en les freqüències que els humans normalment no detectem. Aquesta és l'etapa de compressió amb pèrdua, punt en el qual és injectat un missatge secret. La imatge serà a continuació comprimida sense pèrdua (*lossless*), el que no tindrà cap impacte en la integritat del missatge.

2.4 Utilitzacions modernes de l'esteganografia

Per citar només algunes, podríem parlar d'espionatge industrial, terrorisme i *watermarking* (marques d'aigua).

2.4.1 Espionatge industrial

Els secrets industrials en el món empresarial són un actiu extremadament valuós. Poden revelar informació que podria tenir un efecte extremadament negatiu en els resultats i en la pròpia sostenibilitat d'una companyia. L'espionatge industrial és doncs una amenaça per a qualsevol negoci la vida del qual depengui de la informació. Així per exemple, empleats amb uns privilegis apropiats poden rebre incentius per part de competidors per a revelar informació que podria representar un avantatge pel competidor. És per això que les companyies poden fer tot el possible per a protegir la seva informació mitjançant la implementació de mesures com ara la monitorització de les xarxes digitals, la supervisió del que entra i surt, l'auditoria de les pràctiques d'emmagatzematge de la informació i d'altres mesures de seguretat. Això és el per què l'esteganografia s'ha utilitzat per a ocultar informació secreta en continguts innocus passant així a través de les proteccions que es puguin haver establert.

2.4.2 Terrorisme

Els rumors sobre terroristes utilitzant l'esteganografia van començar al *USA Today* el 5 de febrer de 2001. Els articles, que encara es poden trobar online, es titulaven "*Terrorist instructions hidden online*" i "*Terror groups hide behind Web encryption*". Aquests rumors van ser citats molt sovint, sense mostrar mai cap prova real, especialment després dels atacs de l'onze de setembre. Per cert, aquests articles del *USA Today* van ser escrits per Jack Kelley, qui va ser acomiadat al 2004 per un gran escàndol, donat que sembla ser que havia fabricat moltes històries i s'havia inventat fonts que no existien.

A l'octubre de 2001, el *New York Times* va publicar un article segons el qual al-Qaeda havia utilitzat tècniques esteganogràfiques per a ocultar missatges en imatges, essent després transmeses via e-mail i possiblement via Usenet per a preparar i executar els atacs de l'onze de setembre. Tot i que ha estat descartada per experts en seguretat, la història ha estat àmpliament repetida i torna a sortir de tant en tant. Un manual d'entrenament d'al-Qaeda interceptat no feia cap esment a aquest mètode esteganogràfic. El capítol sobre comunicacions del manual reconeixia la superioritat tècnica dels serveis de seguretat americans, i, en general, recomanava formes de comunicació encoberta de baixa tecnologia. El capítol sobre codis i xifres emfatitzava considerablement la utilització de tintes invisibles en paper de carta tradicional, juntament amb xifres simples com la substitució. L'esteganografia computeritzada amb imatges no hi era esmentada.

En qualsevol cas, es va treballar per a detectar la presència d'informació esteganogràfica en imatges a la web (especialment a eBay, la qual havia estat esmentada a l'article del *New York Times*).

2.4.3 Watermarking

Les marques d'aigua (*watermarks*) digitals no són, estrictament parlant, una forma d'esteganografia, bàsicament pels motius següents:

- No totes les marques d'aigua estan ocultes

- No és el propòsit principal d'una marca d'aigua el ocultar la seva existència

- Un atac exitós contra una marca d'aigua no consisteix en detectar-la, sinó en fer-la inútil, raó per la qual un requeriment addicional del *watermarking* respecte de l'esteganografia és la robustesa contra possibles atacs.

En qualsevol cas, el *watermarking* digital té el seu lloc al camp de l'ocultació de la informació. Les marques d'aigua digitals són utilitzades avui dia principalment com una forma de protecció del *copyright*. Amb l'era digital, va ser evident que les imatges, la música, els vídeos, els llibres i d'altres formes de material amb drets d'autor podien ser clonats perfectament i distribuïts a múltiples receptors sense que els autors o artistes originals rebessin cap mena de pagament o reconeixement. El *watermarking* digital és considerat una forma de prevenir la còpia o modificació il·legal de material amb *copyright*. Les marques d'aigua digitals també s'utilitzen com un mitjà per a impedir la modificació. Sovint són situades a les parts del fitxer més significants des del punt de vista perceptual, de manera que si el fitxer és modificat esdevindrà inservible o com a mínim serà evident que s'hi ha estat "tocant". Una altra forma de *watermarking* és el codi de control del dispositiu que s'ha estat utilitzant amb els DVD i per mitjà del qual el reproductor identificarà la marca d'aigua oculta al disc i que l'indicarà si aquest pot ser copiat una vegada, moltes o cap ni una.

En qualsevol cas, el *watermarking* digital no és la panacea per a protegir materials amb *copyright*, donat que té les seves limitacions i n'hi ha atacs per a eliminar les marques d'aigua digitals (per exemple, l'aplicació StirMark que introdueix distorsions geomètriques bilinears aleatòries per a desincronitzar els algoritmes de *watermarking*; aquest és un atac conegut, l'únic objectiu del qual és eliminar les dades ocultes, no desxifrar-les)

Una tècnica anomenada "*fingerprinting*" és molt similar al *watermarking* donat que oculta quelcom al contingut d'un fitxer. No obstant això, es diferencia del *watermarking* en què no impedeix la còpia. Un número de sèrie o senyal de *copyright* poden ser ocultats de manera transparent en una imatge, una pel·lícula, una cançó, etc., de forma que el productor o l'artista original podrà provar que aquell treball és legítimament seu.

3 Disseny del mètode implementat

Durant la investigació duta a terme per aquest Treball, vaig trobat multitud de mètodes i aplicacions d'esteganografia i *watermarking* per a tot tipus de continguts. Em va cridar, però, l'atenció, el fet que sempre que s'esmenta el mètode de modificació dels bits menys significatius o LSB, se'l considera com a molt fràgil davant de qualsevol modificació del fitxer de cobertura. És aquí on vaig veure la possibilitat d'intentar desenvolupar des de zero un mètode que, sense deixar d'estar basat en la tècnica de la modificació dels bits menys significatius, complís el requisit que es demanava per aquest TFC: que permetés recuperar el missatge ocult malgrat haver modificat "lleugerament" l'objecte que el contingués.

Efectivament, volia desenvolupar un mètode original, en el sentit de què no fos ni una implementació ni tampoc una adaptació d'algun mètode ja existent. Vaig considerar més apropiat, donat que m'enfrontava a tot un Treball Fi de Carrera i no a una prova d'avaluació continuada més de les tantes que he hagut de fer, intentar crear un mètode propi i original, cosa que no semblava gens fàcil donada la omnipresent insistència sobre la fragilitat dels mètodes basats en els LSB.

El sistema implementant funciona sobre fitxers amb imatges de 8, 24 i 32 bits en format BMP, i amb fitxers d'àudio, mono o estèreo, en format WAVE, i està basat, tal i com es pot sospitar després de tot el que s'acaba d'explicar, en la clàssica tècnica de modificació dels LSB. Ja hem avançat també que aquesta tècnica és extremadament sensible a qualsevol modificació del fitxer que contingui el missatge ocult o fitxer de cobertura i, per tant, si no incorporés "quelcom més", no s'estaria satisfent un dels requisits demanats: que es pugui recuperar el missatge ocult malgrat haver modificat lleugerament el fitxer de cobertura quan aquest ja conté el missatge ocult.

És per això que el sistema permet dos modes de funcionament: un que podríem classificar com a fràgil i un altre que es podria classificar com a semi-fràgil.

El sistema d'ocultació fràgil consisteix simplement en agafar el missatge a ocultar (tira de caràcters), encriptar-ho amb la contrasenya si és que aquesta s'ha introduït, i ocultar a la imatge BMP o a l'àudio WAVE de cobertura, mitjançant la modificació dels LSB de determinats bytes, dos bytes que contenen la longitud en bits del missatge ocult i a continuació els bytes formats per la tira de bits en què consisteix el missatge (encriptat o no segons el comentat). Amb aquest sistema, qualsevol canvi a la imatge o a l'àudio pot resultar en una pèrdua total o parcial del missatge ocult (la qual cosa no té per què ser sempre un inconvenient, donat que aquesta tècnica ens serveix per exemple per a comprovar si el fitxer de cobertura ha patit alguna modificació).

I per què es perd el missatge ocult si modifiquem el fitxer que el conté? Doncs l'explicació és molt senzilla: suposem que hem ocultat un missatge a dintre d'una imatge en format BMP. Sabem que al caràcter 'A' li correspon el número binari 01000001 (65 en decimal), i aquests 8 bits estaran distribuïts entre 8 bytes de la imatge de cobertura. Qualsevol canvi en la imatge que modifiqui ni que sigui un sol dels LSB d'aquests 8 bytes, farà que en recuperar el missatge, la concatenació dels 8 bits obtinguts donarà un resultat diferent de l'inicial (01000001), amb la qual cosa el caràcter recuperat mai seria 'A'. I si extrapolem això a tota la tira de caràcters que formen el missatge, entendrem d'on pot venir la pèrdua del missatge ocult original. Per exemple, una inversió dels colors de la imatge BMP de cobertura provocaria la pèrdua total del missatge, donat que si el que s'hi havia ocultat era, per exemple, 01000001, el que es recuperarà podria ser 10111110.

Com podríem solucionar-ho? Com podríem aconseguir la robustesa del sistema enfront de modificacions al fitxer de cobertura? Doncs anem a suposar que, en lloc d'ocultar directament la tira de bits que representen els caràcters del missatge, la qual cosa ja hem vist que és extremadament sensible a qualsevol canvi, ocultem una imatge formada pel missatge. És a dir, si en lloc d'ocultar directament un missatge com a tira de caràcters, generéssim primerament una imatge a la qual es veiés clarament representat el missatge, i després guardéssim aquesta imatge al fitxer BMP o WAVE de cobertura, lleugers canvis al fitxer de cobertura podrien donar lloc a una degradació d'aquesta imatge a la qual s'hi ha representat el missatge, però mentre es pugui llegir el missatge, no hauríem aconseguit el nostre objectiu? Anem a veure un exemple:

- Suposem que el missatge a ocultar és el següent fragment extret de l'inici de 1984, de George Orwell:

  It was a bright cold day in April, and the clocks were striking thirteen. Winston Smith, his chin nuzzled into his breast in an effort to escape the vile wind, slipped quickly through the glass doors of Victory Mansions, though not quickly enough to prevent a swirl of gritty dust from entering along with him.

  The hallway smelt of boiled cabbage and old rag mats. At one end of it a coloured poster, too large for indoor display, had been tacked to the wall. It depicted simply an enormous face, more than a metre wide: the face of a man of about forty-five, with a heavy black moustache and ruggedly handsome features. Winston made for the stairs. It was no use trying the lift.

- L'aplicació ha de ser capaç de generar una imatge, per exemple, en format BMP monocrom, on es pugui llegir aquest missatge:

- A dintre d'aquesta imatge, cada caràcter ve representat per una matriu de 8x8 pixels, amb la qual cosa tenim 64 bits per a representar un caràcter (la imatge és monocromàtica i per tant cada bit representa un pixel). Així doncs, una modificació en un dels bits que formen un caràcter no representaria la pèrdua total del caràcter, cosa que sí passava amb el sistema fràgil anteriorment comentat. És a dir, serà necessari que es vegin afectats bastants bits dels 64 que formen la imatge d'un caràcter per a què aquest deixi de ser reconeixible.

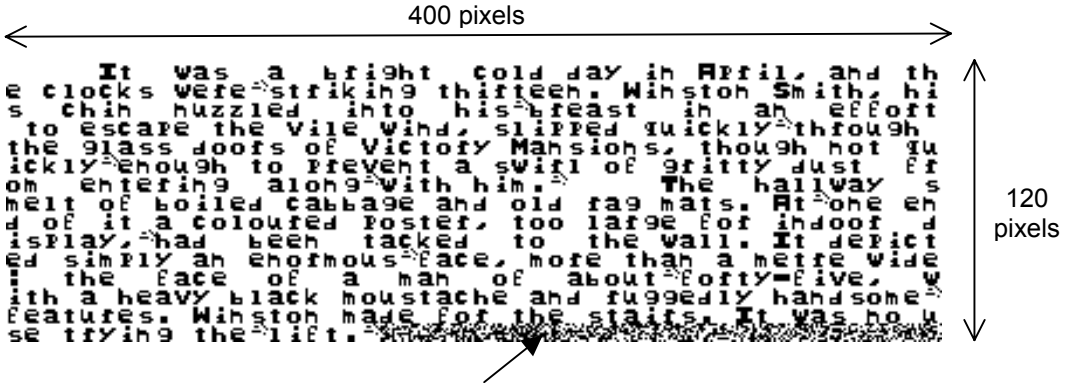
- Un punt important a considerar és que el que s'oculta i recupera no és un fitxer BMP monocromàtic complet, sinó només el seu mapa de bits. És a dir, es considera una capçalera de fitxer BMP "estàndard" (la d'una imatge de mida fixa i monocromàtica), ja que si s'ocultés el fitxer al complet, alteracions en la imatge de cobertura que afectessin els bits de la capçalera podrien fer irrecuperable la imatge. Així per exemple, si el missatge en forma d'imatge s'ha ocultat en una altra imatge, una inversió dels colors d'aquesta última no ens permetria recuperar el fitxer, ja que la capçalera recuperada no correspondria mai a la d'un fitxer BMP. En canvi, utilitzant una capçalera "estàndard", després d'aplicar una inversió de colors a la imatge de cobertura el missatge recuperat en el nostre exemple seria el següent:

Clarament, es pot llegir el mateix missatge que s'havia ocultat. A l'apartat 6 es profunditzarà més en les proves de robustesa.

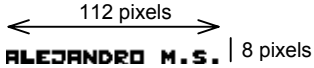
Tal i com s'ha dit, el sistema pot treballar amb fitxers en format WAVE, mono o estèreo, i amb fitxers BMP de 8, 24 o 32 bits. Pel que respecta a aquests darrers, els BMP, treballar amb imatges de 8 bits requereix tenir en compte que estem davant d'imatges indexades. És a dir, amb imatges de 24 o 32 bits, el mapa de bits està format pels bytes que representen cada pixel (3 o 4 bytes respectivament) i és sobre aquests bytes sobre els que es fan les modificacions dels bits menys significatius (LSB). En canvi, en una imatge de 8 bits, el mapa de bits està format per índexs a un mapa de fins a 256 colors, i és sobre aquest mapa sobre el que s'han de fer les modificacions apropiades. Així mateix, a l'hora de treballar amb imatges de 24 o 32 bits, l'aplicació només considera el LSB d'un sol byte dels 3/4 que representen cada pixel (modificant-se per tant només un dels 3 colors que formen el pixel, alternant-se a més a més entre colors per a no estar modificant sempre el mateix). Amb imatges de 8 bits, però, s'ha hagut de permetre la consideració dels LSB de tots els bytes (és a dir, dels 3 colors que formen un pixel), donat que en cas contrari la llargada dels missatges a ocultar quedava excessivament limitada.

Quan es treballa en mode semi-fràgil, és a dir, quan el que s'oculta al fitxer de cobertura no és el missatge directament sinó una imatge BMP a la qual s'hi ha "dibuixat" el missatge, el fet de treballar amb imatges de 24/32 bits o amb imatges de 8 bits implica també diferències a la mida de les imatges amb els missatges que s'hi poden ocultar i, en conseqüència, a la llargada màxima dels missatges. Així ens trobem:

- Fitxer de cobertura de 24/32 bits: excepte per a imatges petites (cosa que ja controla l'aplicació perquè si no hi cap la imatge amb el missatge es llença un missatge d'error), s'hi oculta el mapa de bits d'un BMP monocrom de 400 x 120 pixels (recordem que només s'oculta el mapa de bits, no la capçalera per a evitar que modificacions al fitxer de cobertura deixin irrecuperable la imatge amb el missatge, millorant-se així la robustesa del sistema). Donat que cada caràcter es representa amb 8 x 8 pixels, tenim espai per a 50 caràcters a cada línia (400/8) i un total de 15 línies (120/8), el que proporciona un màxim de 750 caràcters pels missatges a ocultar.



- Fitxer de cobertura de 8 bits: donat que el mapa de colors en té un màxim de 256, i cada color es representa amb 4 bytes (R-G-B + byte reservat), tenim només 1024 LSB "disponibles" (256x4 = 1024 bytes) Com que sabem que cada caràcter es representa amb 64 bits (8 x 8 pixels), tenim 1024/64 = 16 caràcters, però donat que la marca de final de línia de pixels requereix 2 bytes, tenim només espai per a missatges de fins a 14 caràcters.



Pel que fa a la utilització de fitxers WAVE com a fitxers de cobertura, també es consideren únicament els LSB. L'aplicació, en cas d'àudios en estèreo, alterna entre els dos canals a l'hora d'anar modificant els LSB per a evitar que sempre es produeixin les modificacions a un mateix canal. Per descomptat, té en compte de quants bits consta un *sample point* (valor que representa una mostra d'un so en un moment donat)

4 Aspectes concrets de la implementació

Passem ja a analitzar pas per pas el que fa l'aplicació, de nom "Stegano", programada.

El primer que trobem és la definició de tres constants:

- MIDA_MAX_C: estableix la mida màxima pels missatges que es podran ocultar en forma de tira de caràcters (sistema fràgil). Donat que el primer que s'oculta al fitxer de cobertura són dos bytes que indiquen el nombre de bits de què consta el missatge, el número més alt que podem representar amb dos bytes és 65.535 (FFFF en hexadecimal) i donat que el múltiple de 8 més petit o igual que 65.535 és 65.528, 65.528/8 = 8.191 serà el número màxim de caràcters que podrà tenir un missatge.

- MIDA_MAX_I: estableix la mida màxima pels missatges que s'ocultaran representats en una imatge (sistema semi-fràgil), quan el fitxer de cobertura sigui una imatge de 24 o 32 bits, o un fitxer WAVE. Donat que la imatge que es crearà serà de 400 x 120 pixels, i donat que els caràcters s'hi representen amb 8 x 8 pixels, hi cabran 750 (veure apartat 3 anterior per a més detalls)

- MIDA_MAX_I_8: estableix la mida màxima pels missatges que s'ocultaran representats en una imatge quan el fitxer de cobertura és un BMP de 8 bits. La imatge que es crearà serà de 112 x 8 pixels, i a 8 x 8 pixels cada caràcter, hi cabran només 14 (veure apartat 3 anterior per a més detalls)

A continuació trobem la definició d'una sèrie de tipus de dades:

- "byte": donat que es treballa contínuament amb bytes, s'ha considerat convenient definir aquest tipus de dades.

- "fitxerBMP": és una estructura que serveix com a contenidor de les dades dels fitxers BMP que s'han considerat bàsiques pel funcionament del sistema.

- “bmpMonochromeHeader”: ens servirà per a emmagatzemar-hi una capçalera BMP estàndard: la de la imatge monocromàtica a la que s’hi representaran els missatges en el sistema semi-fràgil.

- “formatChunk”, “dataChunk” i “fitxerWave”: tres estructures necessàries per a treballar amb fitxers en format WAVE.

- “caracter”: array amb 8 “bytes” que ens servirà per a contenir la representació monocromàtica d’un caràcter.

Després de les predeclaracions de funcions, trobem ja la funció principal. El primer que fa és la comprovació de què els paràmetres utilitzats siguin correctes. Tal i com es veurà en detall a l’apartat 5 (corresponent al manual d’instal·lació i d’us), la utilització correcta de l’aplicació és mitjançant qualsevol de les dues comandes següents:

- Per a ocultar un missatge en un fitxer de cobertura (un .bmp o un .wav):

```
stegano  c|i [contrasenya] missatge nom_fitxer_origen nom_fitxer_resultat
```

- Per a extreure un missatge d’un fitxer de cobertura:

```
stegano  c|i [contrasenya] nom_fitxer
```

Si les comandes utilitzades no són correctes, l’aplicació s’atura de forma controlada i es llença un missatge que ens recorda quina és la utilització correcta.

Si les comandes utilitzades sí que són correctes, es passa a comprovar de quina forma s’ha introduït el missatge a ocultar. És a dir, si el paràmetre “missatge” és directament la paraula o paraules a ocultar, o si per contra “missatge” és el nom d’un fitxer de text que conté el missatge a ocultar. I com ho fa això? Doncs tracta d’obrir el “presumpte” fitxer de nom indicat pel paràmetre “missatge” i, si aquest no existeix, sabrà que “missatge” conté directament el text a ocultar, mentre que si l’obertura de fitxer ha tingut èxit, sabrà que el missatge a ocultar està contingut a dintre del fitxer en qüestió. En qualsevol dels dos casos, es comprova que el missatge a ocultar no superi la mida màxima en funció del sistema seleccionat mitjançant el primer paràmetre (‘c’, de caràcter, pel sistema fràgil, i ‘i’, d’imatge, pel sistema semi-fràgil).

Si la mida del missatge a ocultar està dintre dels límits, l’aplicació passa a comprovar, en funció del nombre d’arguments utilitzats, quina és l’acció desitjada (en els números d’arguments que s’indiquen a continuació s’hi inclou el que correspon a la crida a l’execució de l’aplicació):

- 5 arguments: ocultació sense encriptació
- 6 arguments: ocultació amb encriptació (l'encriptació es realitza amb una implementació de l'RC4)
- 3 arguments: extracció de missatge no encriptat
- 4 arguments: extracció de missatge encriptat

En cada cas, en funció del sistema seleccionat ('c' o 'i') i del tipus de fitxer de cobertura (BMP o WAVE), es crida la funció apropiada.

- La funció "esteganografiarC" és la funció de partida per a l'ocultació d'un missatge amb el sistema fràgil. El primer que fa és, en funció del tipus de fitxer de cobertura que l'usuari hagi indicat (BMP o WAVE), extreure-hi les dades bàsiques (si el fitxer indicat no és de cap d'aquests formats s'atura l'aplicació i es llença un missatge d'error). L'extracció d'aquestes dades es fa amb la funció que correspongui: "obtenirDadesFitxerBMP" o "obtenirDadesFitxerWave", funcions que, entre d'altres coses, comproven l'existència del fitxer i que efectivament sigui del tipus que indica la seva extensió (és a dir, que un .bmp sigui efectivament un BMP i que un .wav sigui realment un WAVE). Amb les dades obtingudes es comprova si el fitxer de cobertura és un BMP amb compressió, un BMP monocrom o un BMP de 4 bits (fins a 16 colors), casos en els quals s'atura l'aplicació i s'indica que el fitxer de cobertura no és apte. Si el fitxer de cobertura és un BMP de 8, 24 o 32 bits, o bé un WAVE, es controla que la imatge o l'àudio en qüestió sigui suficientment gran per a ocultar-hi el missatge. Per a portar a terme aquest "control de cabuda", l'aplicació té en compte que, per exemple, en un BMP de 24 bits només es considerarà un de cada 3 bytes, o que en el cas d'un WAVE només es considerarà un byte de cada *sample frame* (la mida dels quals ve donada pel camp "wBlockAlign" de l'estructura "formatChunk"; així per exemple, un WAVE estèreo tindrà dos canals i si cada *sample point* és de 16 bits, un *sample frame* ocuparà doncs 4 bytes, 2 bytes per canal)

  A continuació es fa una clonació del fitxer de cobertura, donat que l'ocultació no es farà sobre el fitxer original sinó sobre un clon seu.

  Per últim, crida "ocultaMissatgeEnBMP" o bé "ocultaMissatgeEnWave", en funció del tipus de fitxer de cobertura. Tant la funció "ocultaMissatgeEnBMP" com la funció "ocultaMissatgeEnWave", el primer que fan es guardar a dintre del fitxer de cobertura els 16 bits que componen els 2 bytes que contenen la longitud en bits del missatge a ocultar. A continuació, passen a ocultar la tira de bits que formen el missatge pròpiament dit (el qual estarà encriptat o no segons el comentat inicialment). Consideracions addicionals:

a) “ocultaMissatgeEnBMP”: si el BMP és de 24 o 32 bits, l’ocultació té lloc al mapa de bits, aplicant-se respectivament salts de 4 o 6 bytes per a que cada LSB considerat correspongui a un pixel diferent, anant-se a més a més alternant entre els colors per a no estar sempre modificant només una de les tonalitats. Si el BMP és de 8 bits, l’ocultació no té lloc al mapa de bits, donat que aquest està format per índexs a un mapa de fins a 256 colors; és a dintre d’aquest darrer mapa on té lloc l’ocultació, i aquí sí que es poden modificar tots els bytes que formen els colors per a no limitar excessivament la llargada màxima dels missatges.

b) “ocultaMissatgeEnWave”: l’ocultació té en compte si l’àudio en qüestió és mono o estèreo, donat que en aquest darrer cas va alternant entre canals a l’hora de considerar els LSB per a evitar que totes les modificacions tinguin lloc al mateix canal.

- La funció “esteganografiarI” és la funció de partida per a l’ocultació d’un missatge amb el sistema semi-fràgil. Aquí també el primer que es fa és, en funció del tipus de fitxer de cobertura que l’usuari hagi indicat, extreure-hi les dades bàsiques amb la funció que correspongui: “obtenirDadesFitxerBMP” o “obtenirDadesFitxerWave”. Amb les dades obtingudes es comprova que el fitxer de cobertura no sigui un BMP d’1 o 4 bits, i que el fitxer sigui suficientment gran per a ocultar-hi la imatge monocromàtica que es crearà amb el missatge secret.

  A continuació es crea un fitxer BMP monocrom cridant la funció “crearFitxerBMP”. Aquesta funció crea una imatge BMP de 112 x 8 pixels si el fitxer de cobertura és un BMP de 8 bits, o bé una imatge BMP de 400 x 120 pixels si el fitxer de cobertura és un WAVE o un BMP de 24 o 32 bits. La imatge creada és aleatòria, de manera que aquell espai no utilitzat (per ser el missatge inferior a 14 caràcters en el primer cas o a 750 en el segon) no serà tot blanc o tot negre (amb la qual cosa la seva representació binària no serà tot zeros o tot uns, la qual cosa faria molt més fàcil la detecció de si hi ha o no un missatge ocult en el fitxer de cobertura per part d’un possible atacant).

  El fitxer BMP creat, de nom “missatgeBMP.bmp”, és utilitzat a continuació per a dibuixar-hi el missatge secret mitjançant la funció “escriureMissatgeEnFitxerBMP”. Els caràcters que pot contenir un missatge són aquells que tenen un codi ASCII entre el 32 i 126 (a dintre del codi font de la funció “escriureMissatgeEnFitxerBMP” es pot veure quins són aquests caràcters)

A continuació es clona el fitxer de cobertura per a treballar sobre el clon i no sobre el fitxer original, es llegeix el mapa de bits del fitxer BMP que conté la representació del missatge i s'encripta o no en funció de si s'ha utilitzat una contrasenya en la crida a l'aplicació. Aquest mapa de bits, encriptat o no segons sigui el cas, és ocultat finalment al clon del fitxer de cobertura. S'ha d'insistir en el fet de què el que s'oculta al fitxer de cobertura no és pas el fitxer BMP complet amb el missatge, sinó només el seu mapa de bits. D'aquesta manera s'evita que possibles modificacions del fitxer que conté el missatge secret deixin irrecuperable el missatge per afectar bits de la seva capçalera, amb la qual cosa el que s'extrauria no correspondria a un fitxer BMP correcte. Per altra banda, tal i com es feia en el cas del sistema fràgil, aquí també es va alternant entre colors si el fitxer és un BMP o entre canals si és un WAVE estèreo.

- Les funcions "extreureCdeBMP" i "extreureCdeWave" són les que s'utilitzen en el sistema fràgil per a l'extracció de missatges ocults en fitxers BMP i WAVE respectivament, essent el missatge extret mostrat per pantalla.

- Les funcions "extreureIdeBMP" i "extreureIdeWave" són les utilitzades al sistema semi-fràgil per a extreure els missatges ocults, en forma de mapa de bits corresponents a imatges BMP monocromàtiques, de fitxers BMP i WAVE respectivament, guardant-se el resultat de l'extracció en el mapa de bits d'un fitxer de nom "missatgeExtret.bmp", el qual és creat prèviament amb la corresponent capçalera estàndard.

5 Manual d'instal·lació i d'ús

L'aplicació no requereix cap mena d'instal·lació. S'adjunta el codi font, així com una compilació feta sobre Windows 98 amb el Bloodshed Dev-C++ (versió 4).

Pel que fa a la seva utilització, els paràmetres a utilitzar (en l'ordre que s'exposa) són els següents:

Per a ocultar un missatge en un fitxer de cobertura (un .bmp o un .wav):

```
stegano c|i [contrasenya] missatge nom_fitxer_origen nom_fitxer_resultat
```

1. **c | i** : si es vol guardar el missatge com a tira de caràcters (sistema "fràgil") s'utilitzarà 'c', mentre que si es vol guardar el missatge com a imatge (sistema "semi-fràgil") s'utilitzarà 'i'. A l'hora de recuperar el missatge s'haurà d'escollir també l'opció apropiada ('c' o 'i') en funció de quin sistema d'ocultació es va emprar.

2. **[contrasenya]** : la utilització de contrasenya és voluntària. Si s'utilitza, s'aplicarà encriptació, amb una implementació de l'RC4, a les dades abans de guardar-les a la imatge o àudio de cobertura.
3. **missatge**: es tracta del missatge a ocultar. Pot ser tant el text del missatge (entre " " si té més d'una paraula) com el nom d'un fitxer en format .txt que contingui el missatge a ocultar.
4. **nom_fitxer_origen**: nom del fitxer BMP (funciona amb BMPs de 8, 24 i 32 bits) o del fitxer WAVE (pot ser mono o estèreo) a utilitzar com a fitxer de cobertura.
5. **nom_fitxer_resultat**: nom que es vol que tingui el fitxer BMP o WAVE resultant.

Per a extreure un missatge d'un fitxer de cobertura:

```
stegano c|i [contrasenya] nom_fitxer
```

1. **c | i** : s'utilitzarà 'c' si el missatge es va ocultar com a tira de caràcters, o 'i' si es va ocultar en forma d'imatge.

2. **[contrasenya]** : només serà necessària si el missatge es va encriptar abans d'ocultar-lo.

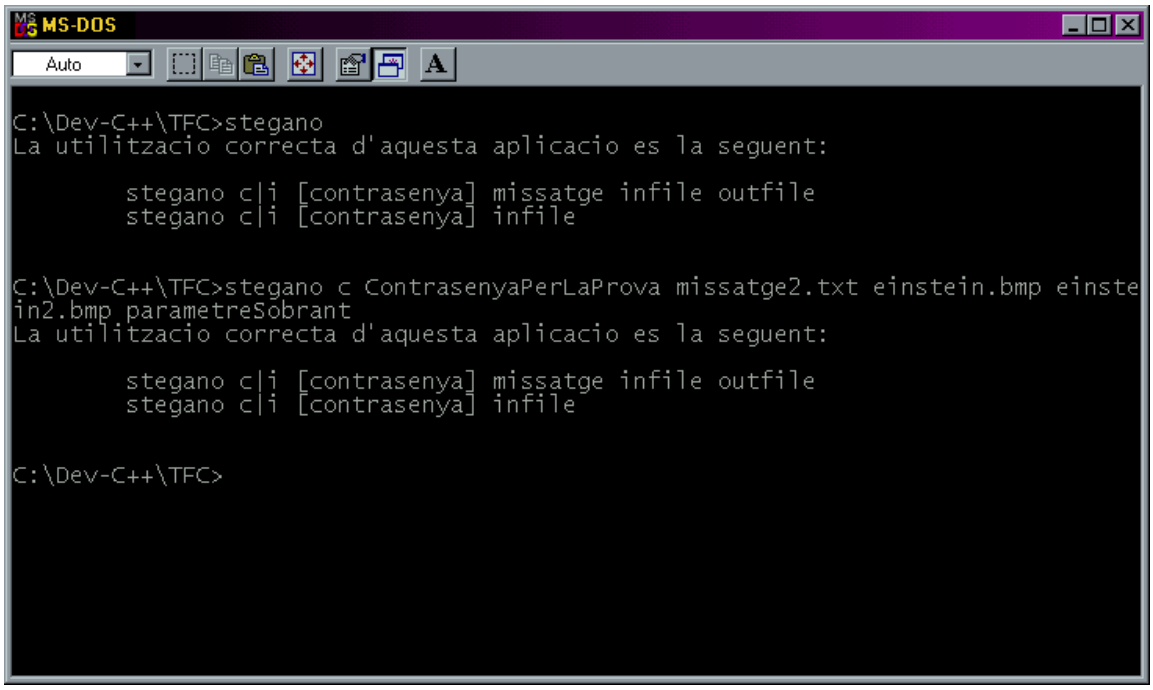
3. **nom_fitxer**: nom del fitxer BMP o WAVE del qual es vol extreure el missatge.

6 Descripció de les proves

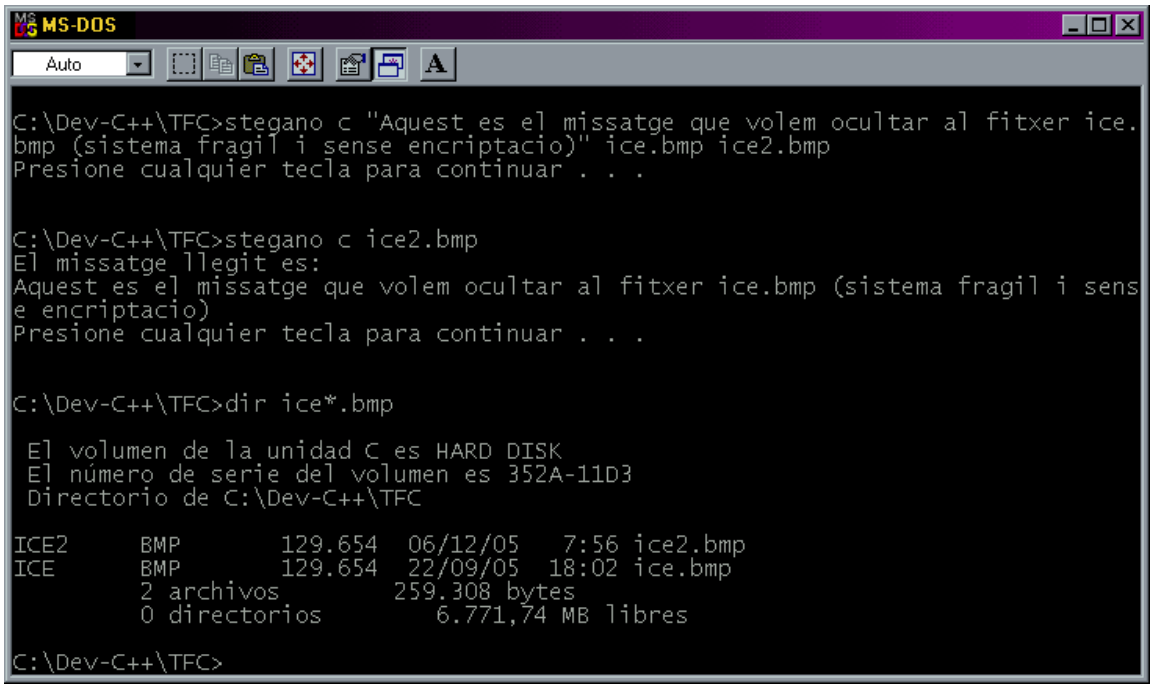
Es farà en primer lloc una sèrie de proves de funcionament, per a passar a continuació a fer les proves de robustesa.

6.1 Proves de funcionament

Cridem primerament l'aplicació sense paràmetres i després amb un número de paràmetres erroni per a comprovar l'aturada controlada de l'aplicació i el llançament dels missatges d'error:



Anem a ocultar ara, sense encriptació i amb el sistema fràgil (opció 'c'), un missatge, introduït per la línia de comandes, al fitxer "ice.bmp"; tot seguit es fa l'extracció i es fa un dir per a comprovar que tant el fitxer de cobertura original com el resultant ("ice2.bmp") tinguin exactament la mateixa mida:



A continuació podem veure la imatge original i la que conté el missatge secret:

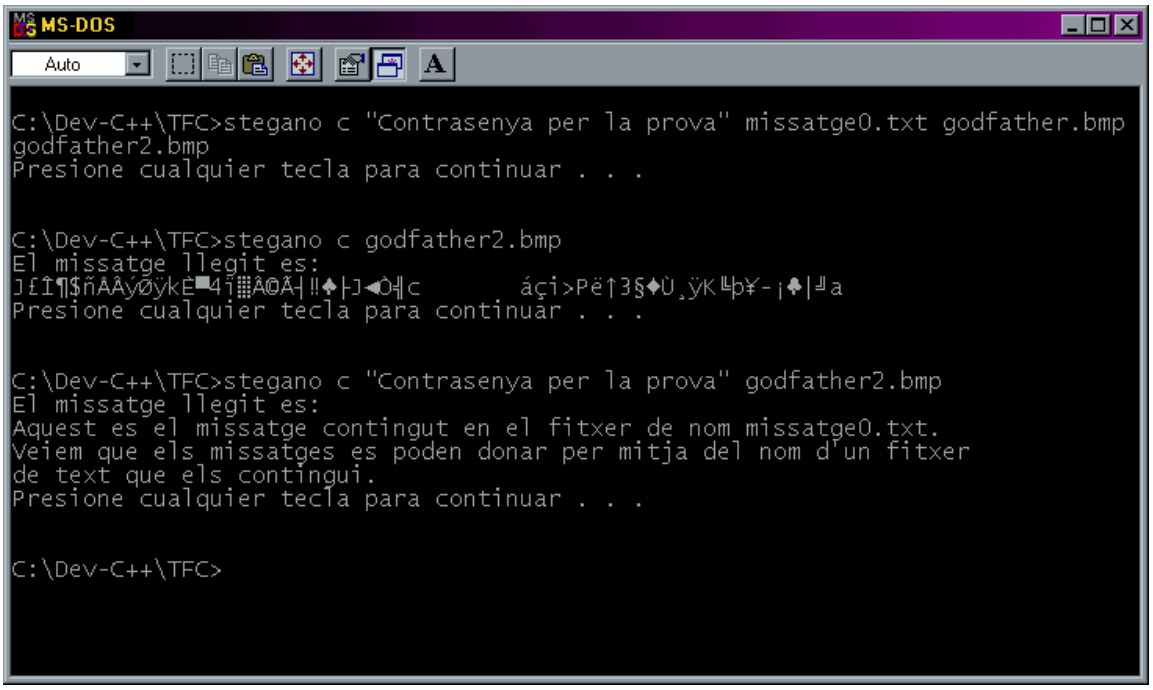
Imatge original

Imatge resultant

Com es pot comprovar, visualment no es pot percebre cap diferència entre totes dues imatges (es poden per exemple ampliar tot el que es vulgui per a intentar trobar diferències de tonalitats entre pixels), i ja hem vist que tots dos fitxers tenen exactament la mateixa mida.

Provem ara a ocultar un missatge, contingut en el fitxer “missatge0.txt”, i amb encriptació. A continuació provem d’extreure el missatge sense contrasenya per a veure què obtenim i finalment ho tornem a provar però ja amb la contrasenya correcta. El contingut del fitxer “missatge0.txt” és el següent:

> “Aquest es el missatge contingut en el fitxer de nom missatge0.txt.
> Veiem que els missatges es poden donar per mitja del nom d'un fitxer
> de text que els contingui.”

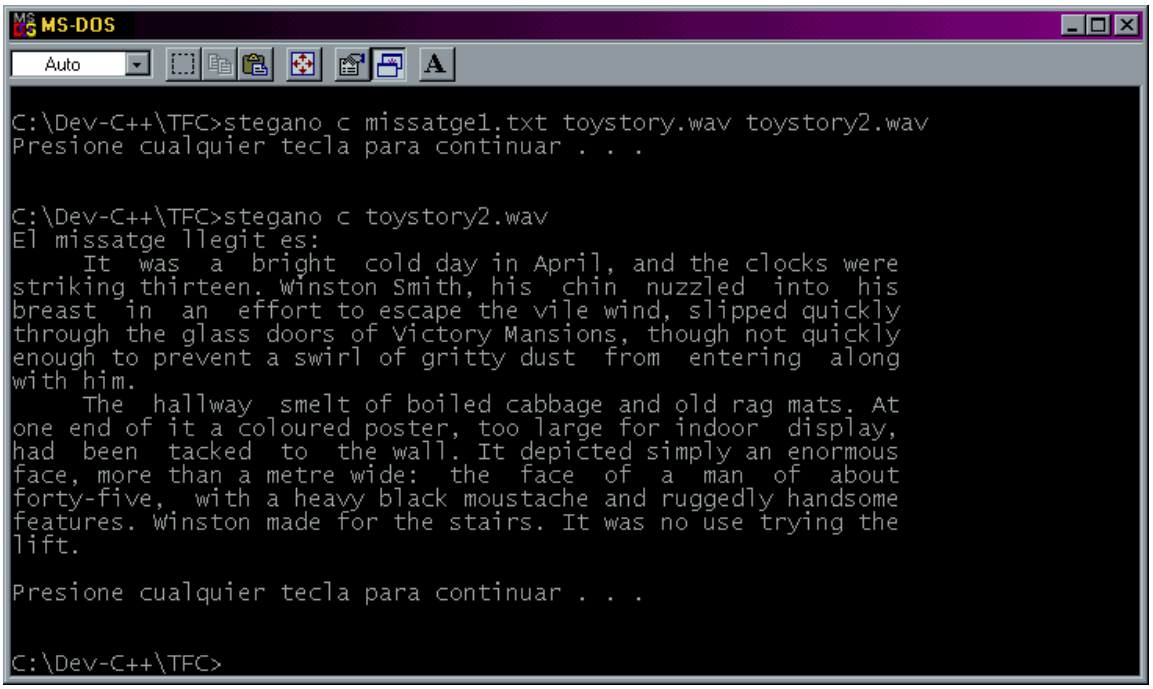


Veiem a continuació tant la imatge original com la resultant (tots dos fitxers són de 404.654 bytes):

Imatge original

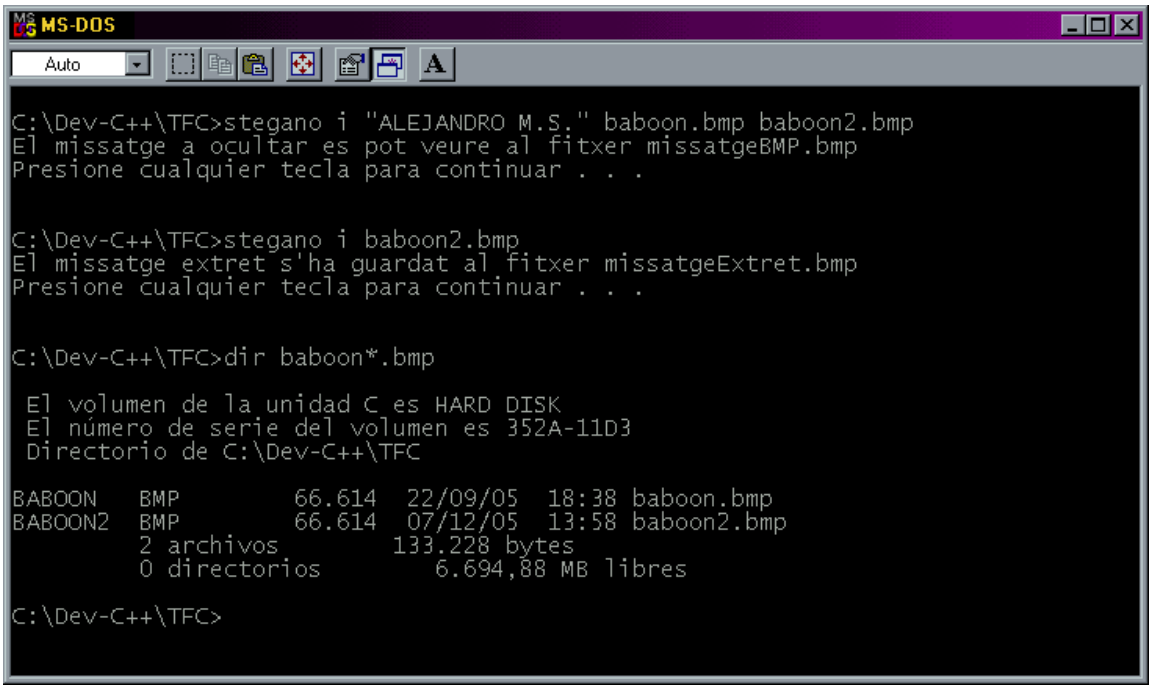
Imatge resultant

Provem ara a ocultar el missatge contingut al fitxer “missatge1.txt” (un extracte del començament de 1984 de George Orwell que ja ha aparegut anteriorment) en un fitxer WAVE:



Tots dos fitxers, original y resultant, són de 160.342 bytes i no és pot percebre cap diferència en escoltar-los. És cert que aquest àudio en concret té un soroll de fons i pot donar a entendre que s’ha utilitzat expressament per aquesta circumstància: res més lluny de la meva intenció. El “problema” és que en passar qualsevol cançó a WAVE, la mida del fitxer és excessiva i per tant impedeix d’adjuntar-la al lliurament d’aquest TFC i per això s’ha utilitzat un fitxer relativament petit. En qualsevol cas, es pot provar amb el fitxer WAVE que es desitgi i es veurà que realment no es pot percebre cap diferència.

Ocultem ara un missatge amb el sistema semi-fràgil en un fitxer BMP de 8 bits (la mida màxima dels missatges per aquest tipus de fitxer de cobertura és de 14 caràcters) i a continuació s’extreu:



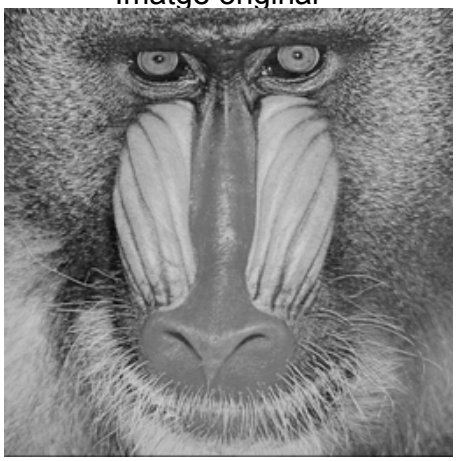
L'aplicació ens informa de què el missatge a ocultar es pot veure al fitxer "missatgeBMP.bmp" (això serveix per a què qui envia el missatge pugui comprovar què és el que s'oculta i per tant què és el que podrà extraure el destinatari) i de què el missatge extret s'ha guardat al fitxer "missatgeExtret.bmp". Anem a veure'ls:

Missatge original: ALEJANDRO M.S.

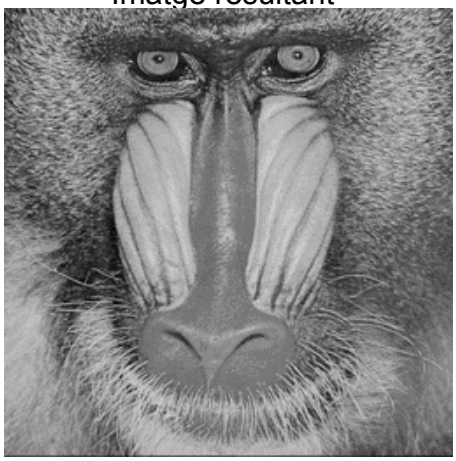
Missatge extret: ALEJANDRO M.S.

Comprovem en qualsevol cas que tant la imatge de cobertura original com la resultant siguin iguals (tot i que en realitat sabem que iguals exactament no són):

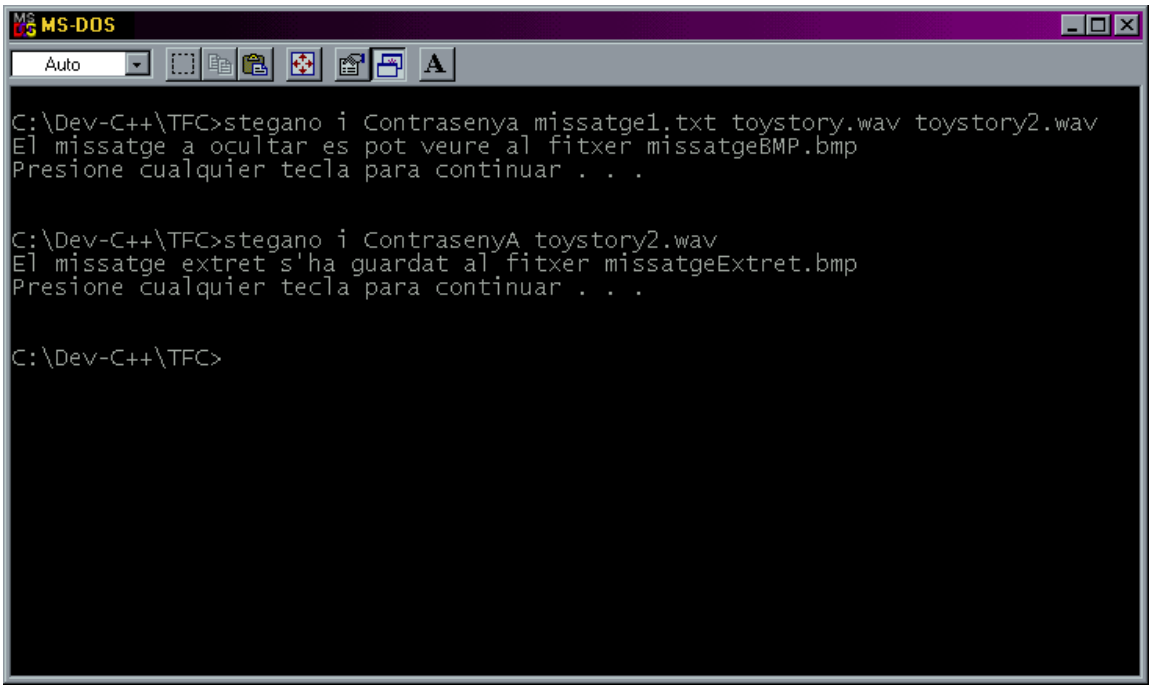
Imatge original



Imatge resultant

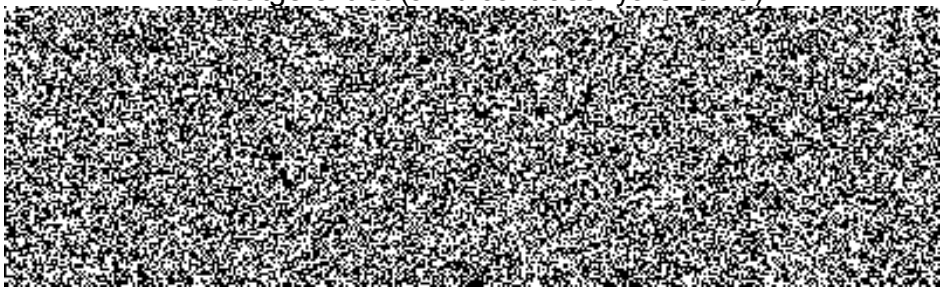


Ocultem ara amb contrasenya i amb el sistema semi-fràgil el missatge contingut al fitxer “missatge1.txt” en un fitxer WAVE. A continuació extraiem el missatge ocult però equivocant intencionadament la darrera lletra de la contrasenya per a veure què és el que s’extreu:

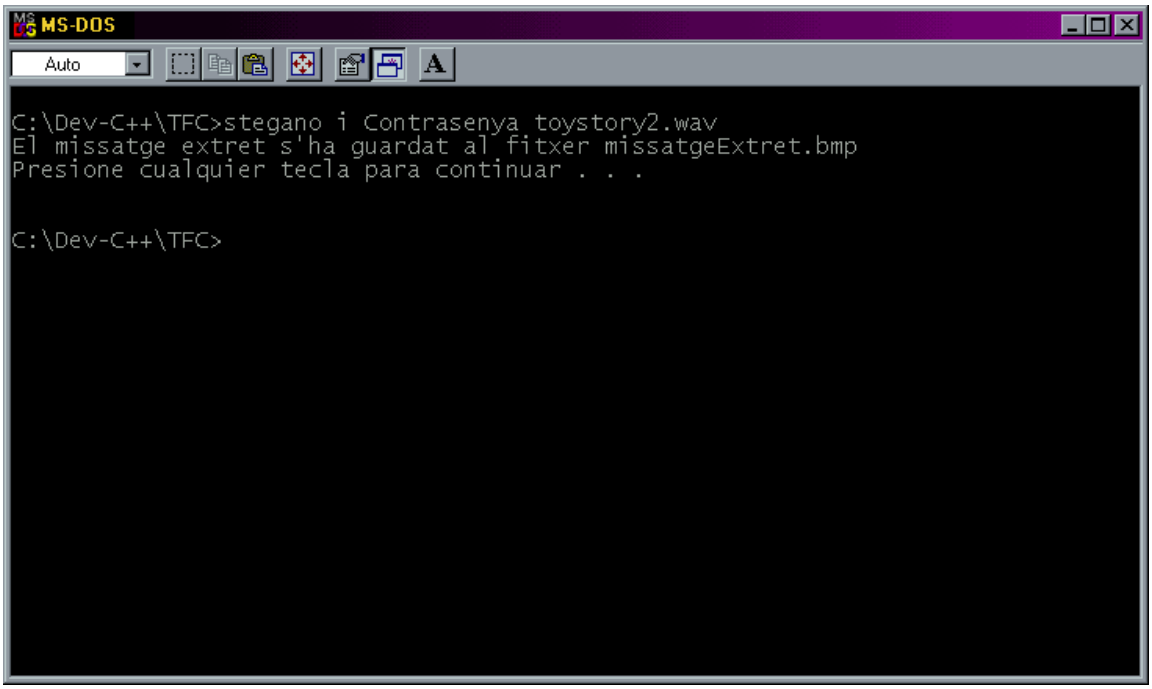


Missatge ocultat

Missatge extret (amb contrasenya errònia)



Sembla doncs que l'encriptació funciona correctament. Provem ara l'extracció amb la contrasenya correcta:

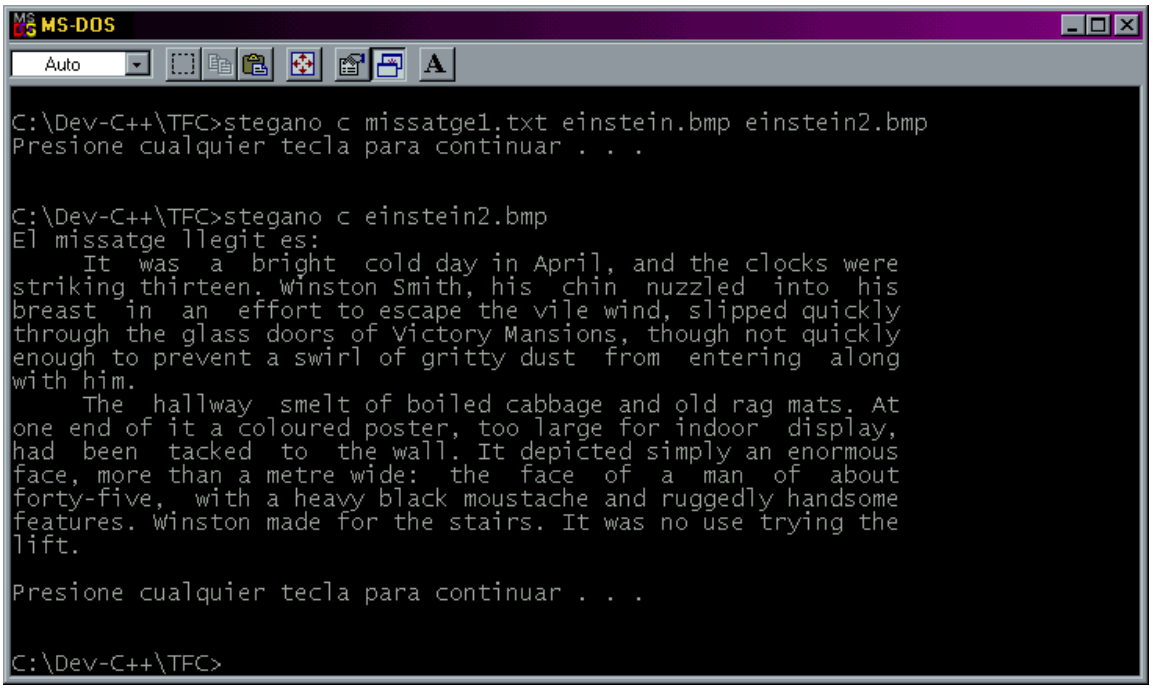


El missatge extret és ara:

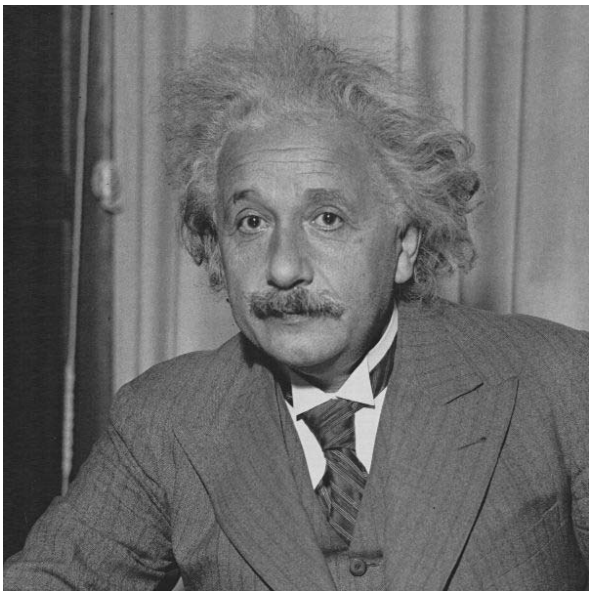
6.2 Proves de robustesa

Anem ara a comprovar que el sistema implementat compleixi amb el requisit establert a l'enunciat d'aquest TFC: que permeti recuperar el missatge ocult malgrat haver modificat "lleugerament" l'objecte. Per a fer les proves, utilitzarem només fitxers de cobertura BMP, donat que resulta més fàcil mostrar aquí les modificacions fetes en una imatge que no pas en un àudio.

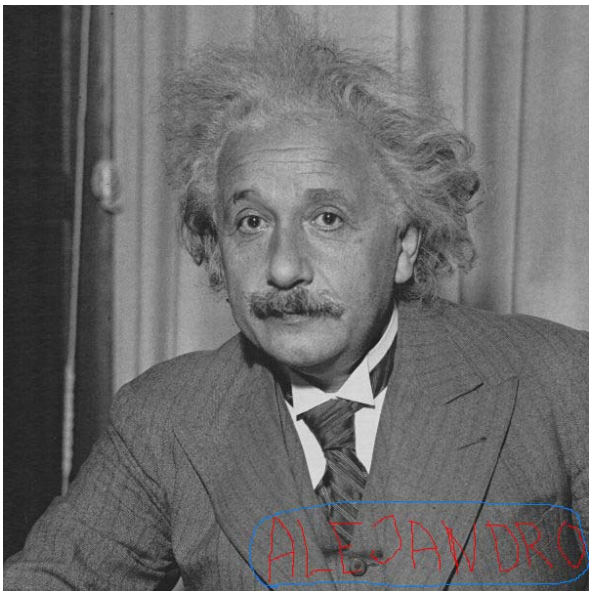
En primer lloc, ocultem de nou el missatge contingut al fitxer missatge1.txt utilitzant el sistema fràgil, i sense haver fet cap modificació al fitxer de cobertura resultant, l'extraiem:



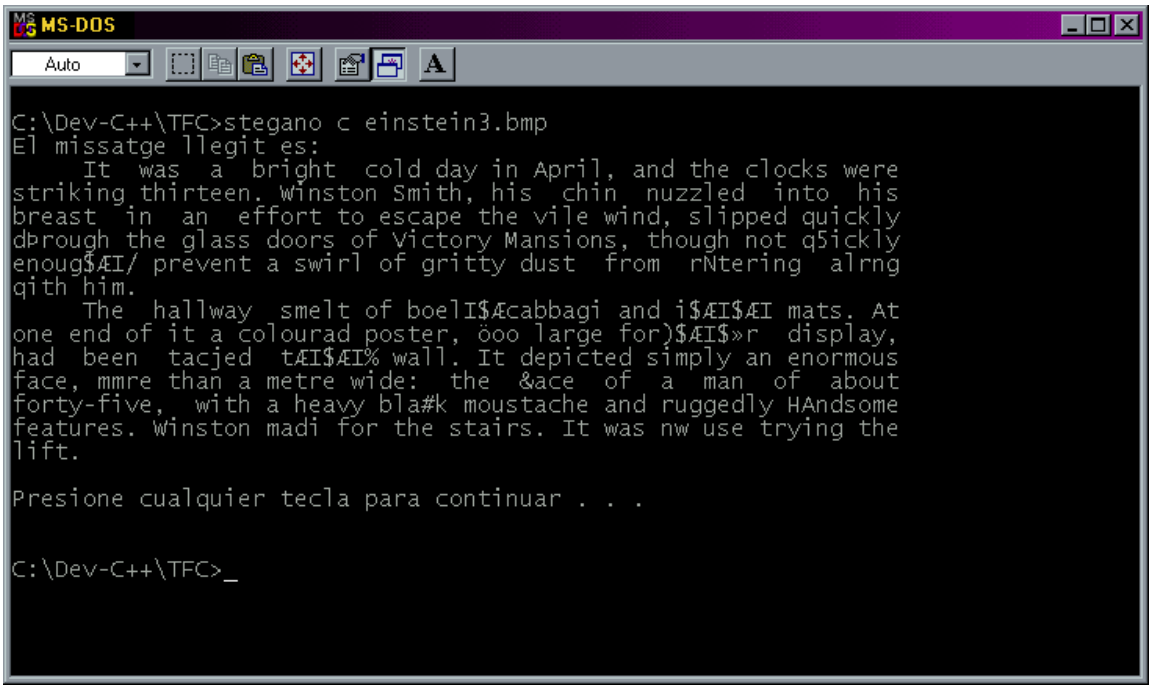
La imatge continguda en el fitxer "einstein2.bmp" i que per tant conté ja el missatge ocult és la següent:



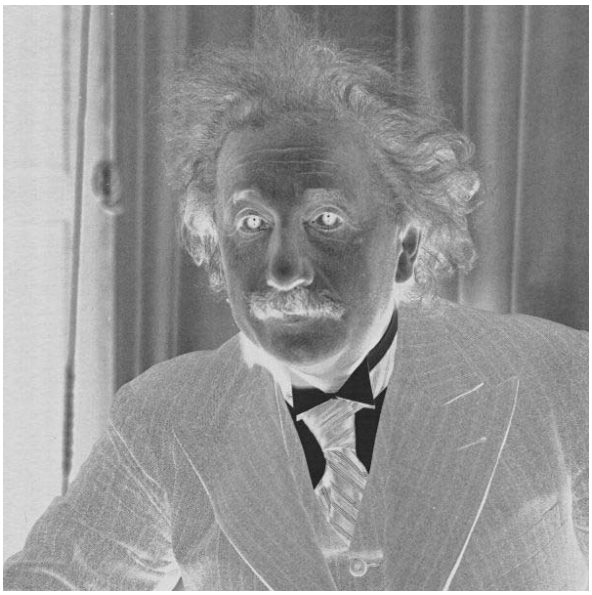
Obrim aquest fitxer, per exemple amb el programa Paint que ve amb el propi Windows i el guardem amb el nom “einstein3.bmp”. Anem a fer alguna bretolada amb aquesta imatge com per exemple, signar-la a la part inferior dreta:



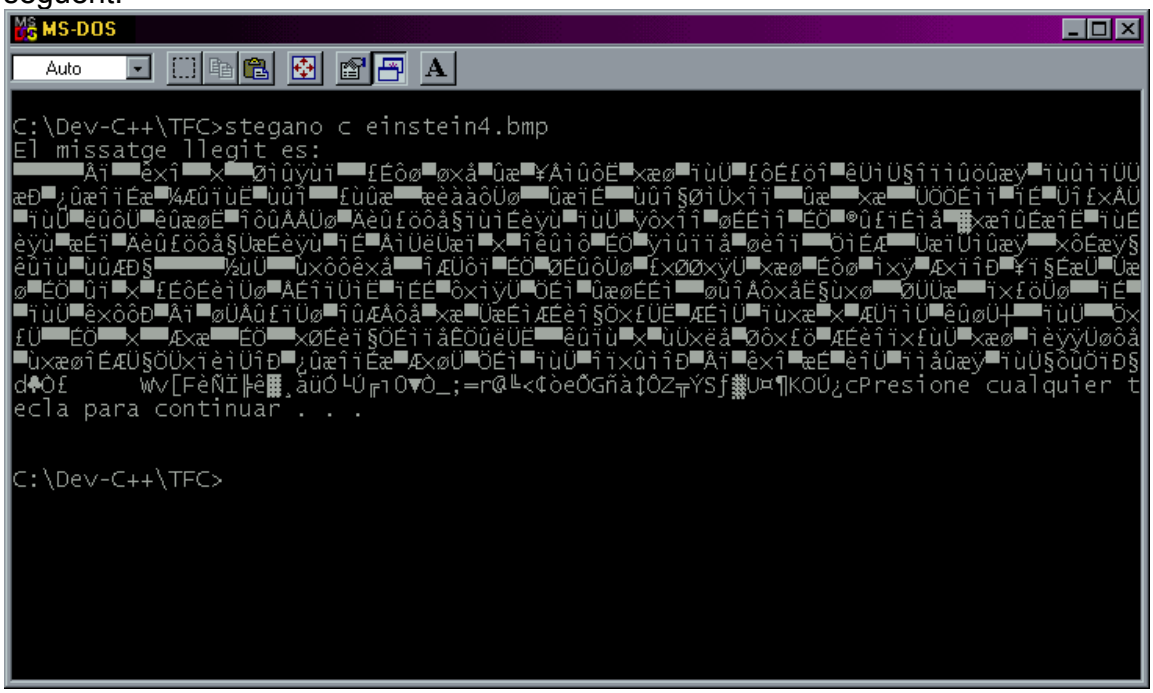
Si ara extraiem el missatge ocult d’aquest fitxer “einstein3.bmp”, obtenim el següent resultat:



Com podem comprovar, s'han vist modificats bastants caràcters, fent que algunes paraules resultin inintel·ligibles. Però què passa si, partint de nou del fitxer "einstein2.bmp" fem una modificació més radical com per exemple una inversió de colors i guardem la imatge resultant al fitxer "einstein4.bmp":



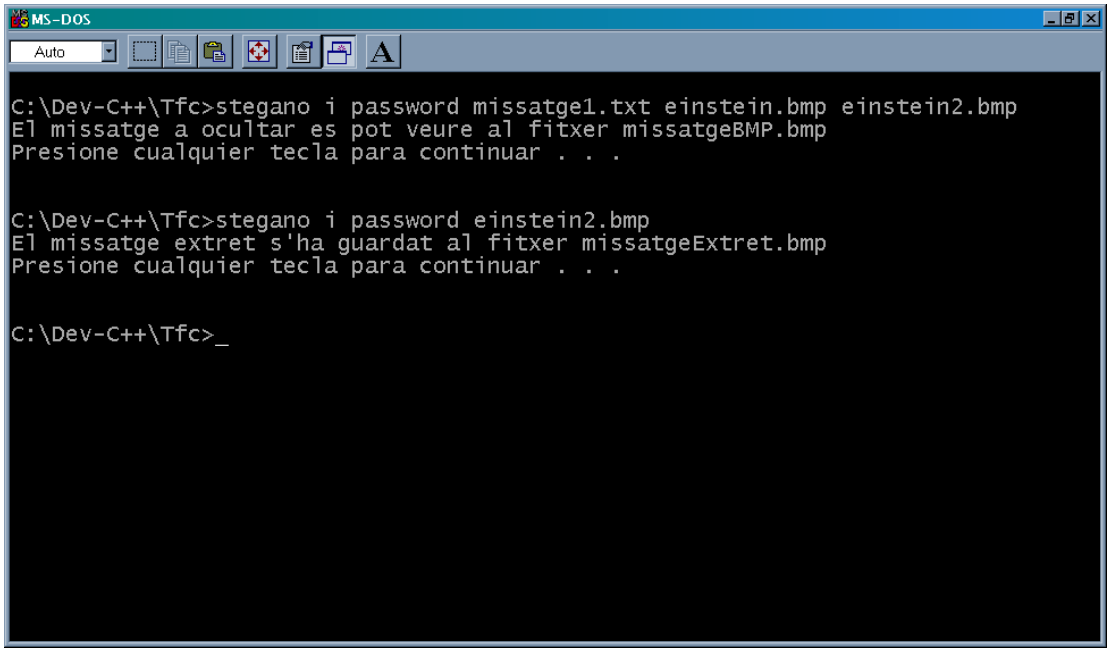
Si ara extraiem el missatge ocult d'aquest fitxer "einstein4.bmp" obtenim el següent:



Ara sí que hem perdut completament el missatge que s'hi havia ocultat. És cert que sempre podem tornar a invertir els colors amb el que obtindríem de nou la imatge inicial, d'on extrauríem el missatge correcte, però això és només a efectes il·lustratius del que una modificació del fitxer de cobertura pot implicar amb aquest sistema fràgil. En qualsevol cas, ja érem conscients de la fragilitat d'aquest sistema, d'aquí el seu qualificatiu, i és que donat que un caràcter, en equivaler biunívocament a un valor enter representable amb 8 bits, es perdrà si modifiquem ni que sigui un sol dels 8 bits que representen el seu valor enter.

Passem doncs a provar el sistema semi-fràgil. Recordem que el principi en el qual es basa aquest sistema és que, en representar un caràcter amb una imatge monocromàtica de 8 x 8 pixels i per tant amb 64 bits, les modificacions al fitxer de cobertura han d'afectar una part considerable d'aquests 64 bits per a què el caràcter en qüestió deixi de ser identificable.

Tornem a guardar el missatge contingut a "missatge1.txt" al fitxer de cobertura "einstein.bmp":



Abans de fer cap modificació, comprovem el contingut de "missatgeBMP.bmp" (missatge que s'oculta) i "missatgeExtret.bmp" (missatge que es recupera):

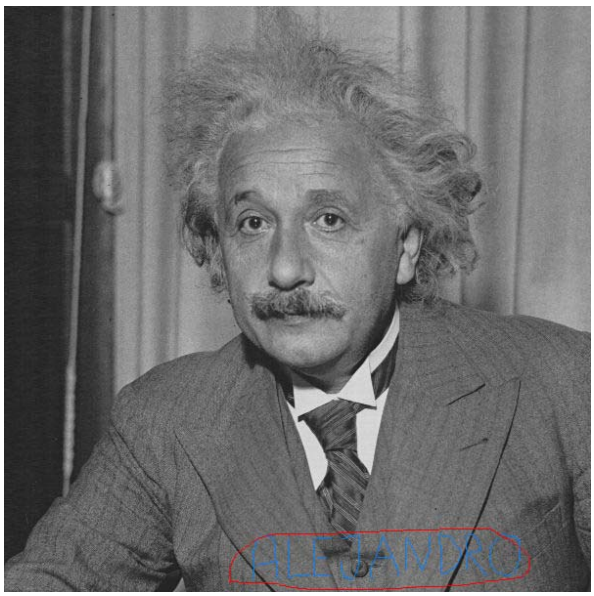
missatgeBMP.bmp

It was a bright cold day in April, and th
e clocks were striking thirteen. Winston Smith, hi
s chin nuzzled into his breast in an effort
to escape the vile wind, slipped quickly through
the glass doors of Victory Mansions, though not qu
ickly enough to prevent a swirl of gritty dust ef
om entering along with him. The hallway s
melt of boiled cabbage and old rag mats. At one en
d of it a coloured poster, too large for indoor d
isplay, had been tacked to the wall. It depict
ed simply an enormous face, more than a metre wide
: the face of a man of about forty-five, w
ith a heavy black moustache and ruggedly handsome
features. Winston made for the stairs. It was no u
se trying the lift.

missatgeExtret.bmp

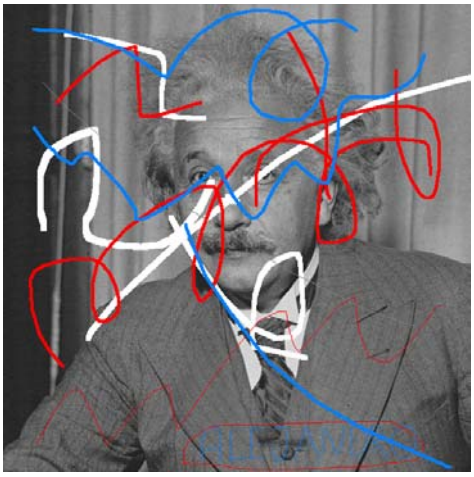
It was a bright cold day in April, and th
e clocks were striking thirteen. Winston Smith, hi
s chin nuzzled into his breast in an effort
to escape the vile wind, slipped quickly through
the glass doors of Victory Mansions, though not qu
ickly enough to prevent a swirl of gritty dust ef
om entering along with him. The hallway s
melt of boiled cabbage and old rag mats. At one en
d of it a coloured poster, too large for indoor d
isplay, had been tacked to the wall. It depict
ed simply an enormous face, more than a metre wide
: the face of a man of about forty-five, w
ith a heavy black moustache and ruggedly handsome
features. Winston made for the stairs. It was no u
se trying the lift.

Agafem ara el fitxer “einstein2.bmp” contenidor del missatge ocult i tornem a fer la bretolada d’espatllar la imatge amb una signatura:



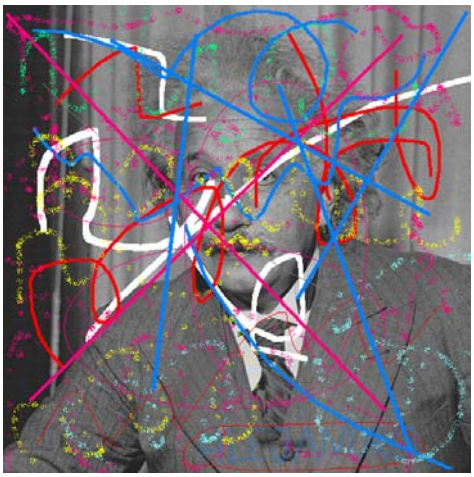
Extraiem ara el missatge contingut en aquesta imatge modificada:

Podem veure com alguns pixels han canviat de blanc a negre i viceversa, però el missatge resulta perfectament intel·ligible. Anem a fer-hi algunes modificacions més:



Ara sí que la imatge ha sigut modificada de manera més agressiva. Comprovem què podem extreure de la mateixa:

Doncs sembla ser que amb aquest sistema ens costarà una mica més de fer malbé el missatge. Continuem insistint:



Què ens queda ara del missatge ocult?

Doncs encara resisteix, tot i que les modificacions introduïdes a la imatge de cobertura l'han deixada totalment inservible.

Si qui ha d'enviar el missatge secret sospita que el fitxer de cobertura pot ser modificat, a més a més d'utilitzar aquest sistema semi-fràgil, podria utilitzar només caràcters en majúscules. I això per què? Doncs pel fet de què les lletres majúscules es representen amb un major nombre de pixels en negre dels 64 possibles, amb el que les modificacions a la imatge de cobertura han de ser encara més grans per a "deformar" totalment un caràcter. Anem a comprovar-ho amb el missatge anterior, però ara en majúscules ("missatge3.txt) i amb un altre fitxer de cobertura.

Imatge original

Imatge resultant

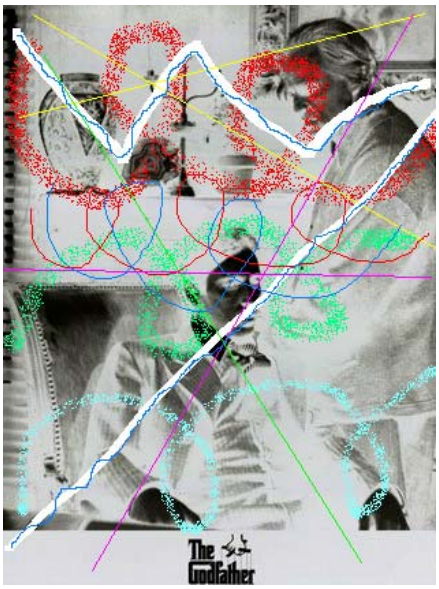
missatgeBMP.bmp (missatge ocultat)

missatgeExtret.bmp (missatge recuperat)

Fem per començar un canvi que ja vam veure que amb el sistema fràgil provocava la pèrdua total del missatge ocult: una inversió de colors.

Recuperem el missatge ocult d'aquesta imatge modificada per a comprovar que efectivament aquesta modificació no ens ha fet perdre el text:

Fem ara algunes modificacions importants sobre la imatge:



Què ens queda ara del missatge ocult:

Sembla evident que el missatge és perfectament intel·ligible, i això tot i que les modificacions fetes a la imatge són quelcom més que lleugeres.

7 Comentaris i conclusions

Fins aquí l'anàlisi del sistema implementat així com les proves de la seva robustesa. Sembla evident que amb el sistema semi-fràgil l'aplicació és robusta enfront de cert tipus de modificacions dels fitxers de cobertura, com per exemple i pel cas d'imatges, la inversió de colors i el dibuixar-hi a sobre.

S'ha especificat des de bon principi que l'objectiu d'aquest TFC era desenvolupar des de zero un sistema d'esteganografia que, sense deixar d'estar basat en la tècnica de modificació dels bits menys significatius, fos robust. És cert que el sistema implementat no és robust enfront de modificacions com ara la transició entre formats BMP i JPEG per les imatges, o entre WAVE i MP3 pels àudios, però aquest no ha estat mai el seu objectiu. Per a fer això només hauria d'haver implementat algun dels algorismes que es poden trobar a la xarxa, i dels quals fins i tot un pot trobar el codis font, i adaptar-lo a les meves necessitats i/o gustos. Però com ja s'ha dit en un altre punt d'aquest Treball, vaig creure més oportú desenvolupar un mètode original i per tant innovador. Espero haver complert el meu objectiu.

8 Bibliografia i recursos usats

Com a recursos, l'únic que he utilitzat és el compilador Bloodshed Dev-C++, versió 4, (http://www.bloodshed.net/devcpp.html), havent programat l'aplicació en C.

Pel que fa a les fonts d'informació, totes han vingut d'Internet:

- Digital Watermarking World:

  http://www.watermarkingworld.org/

- Steganography & Digital Watermarking:

  http://www.jjtc.com/Steganography/

- The Information Hiding Homepage. Digital Watermarking & Steganography

  http://www.petitcolas.net/fabien/steganography/

- Stegoarchive.com

  http://www.stegoarchive.com/

- Articles de Corinna John:

  http://www.codeproject.com/script/articles/list_articles.asp?userid=475133

- Format dels fitxers BMP:

  http://www.wotsit.org/download.asp?f=bmp

- Format dels fitxers WAVE:

  http://www.borg.com/~jglatt/tech/wave.htm