Esteganografia en àudio

Estudiant

Miquel Parejo Cano (ETIS)

Consultor

Antoni Martínez Ballesté

15 de Juny de 2006

Taula de contingut

1. Fonaments

Què és l'esteganografia?

Essencialment, l'esteganografia és l'art d'ocultar informació privada o delicada en un contenidor que aparentment sembla innocu. La paraula esteganografia prové del grec *steganós* (amagat) i *graptos* (escriptura). Explicat d'una manera planera, si hom presenciés la informació no hauria d'apreciar res que fos susceptible de ser analitzat ja que aquesta informació no sembla que pugui contenir res de valuós. L'esteganografia ha estat utilitzada des de fa molts anys per transportar informació de manera que no pogués ser detectada fins que arribés a la seva destinació. Així doncs, és competència del remitent i del destinatari establir un mètode comú i per tant es requereix d'un mínim de comunicació prèvia.

S'acostuma a confondre esteganografia amb criptologia. Tot i que tots dos coexisteixen en un mateix marc, no son el mateix. La seva missió es la de protegir la informació; l'esteganografia ho fa intentant ocultar-la de manera que passi desapercebuda i la criptografia l'encripta de tal manera que tot i hom sàpiga que hi ha informació sensible no és capaç d'interpretar-la.

La percepció humana

L'esteganografia es basa en el fet que els sentits humans son sensiblement menys capaços d'analitzar informació si els comparem amb aquells realitzats per màquines o fins i tot d'altres animals de la terra. La vista o la oïda són incapaços de detectar canvis menors i subtils en qualsevol representació auditiva o visual, fent de l'esteganografia una manera efectiva d'ocultar informació privada.

Breu història de l'esteganografia

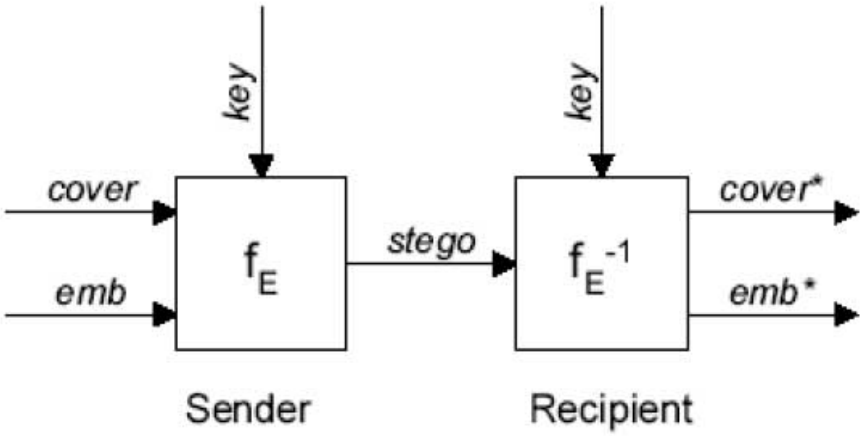
Durant la història de la humanitat s'ha donat multitud de situacions en les quals transportar informació de manera oculta i indetectable era una qüestió

en ocasions vital. A continuació s'exposen alguns dels exemples més rellevants que poden donar a entendre la importància real d'aquest art:´

- A l'Antiga Grècia utilitzaven un mètode molt bàsic per transportar informació. S'escollia una persona que s'utilitzaria de missatgera i se li afaitava el cap. El text secret era literalment tatuat al seu cap i s'esperava fins que el cabell tornava a créixer. Aleshores era el moment que el missatger emprengués la marxa. En cas que fos capturat era molt difícil que les tropes enemigues suposessin que aquella persona duia un missatge de text tatuat al cap. Aquí ja veiem la fortalesa de l'esteganografia: el missatge passava desapercebut. El principal inconvenient d'aquesta tècnica era que una persona no podia ser missatgera dues vegades. Això es va solventar posteriorment utilitzant tintes que si es podien esborrar.
- Durant la Segona Guerra Mundial es va utilitzar tintes invisibles per amagar informació en cartes aparentment innòcues. Era comú utilitzar llet o suc de llimona com a tinta invisible doncs tenen la propietat que un cop escalfades s'enfosqueixen, revelant així el missatge secret.

Com funciona?

Avui en dia hi ha multitud de maneres d'amagar informació dins d'arxius de música o imatges, per exemple. Aquests arxius acostumen a tenir informació informació irrellevant i/o redundant que pot ser substituïda per missatges ocults. La figura següent mostra el mètode bàsic del funcionament de l'esteganografia (extreta del Segon Taller en ocultació d'Informació de l'Abril de 1998 a Portland, Estats Units):



f_E: steganographic function "embedding"
f_E^{-1}: steganographic function "extracting"
cover: coverdata in which *emb* will be hidden
emb: message to be embedded
key: parameter of f_E
stego: coverdata with embedded message

Figura 1

Cover és la imatge, vídeo o so original. **Emb** és el missatge secret. **Key** és el paràmetre que controla tot el procés d'ocultació i **stego** és l'arxiu resultat que conté la informació inicial i el missatge ocult.

Actualment hi ha tres tècniques per a tal menester: substitució, addició i generació d'arxius nous.

- La tècnica de substitució consisteix en cercar totes aquelles dades dins del *continent* original que no son necessàries bé perquè son redundants bé perquè son irrellevants. Un dels mètodes més empleats en aquest camp és l'anomenat LSB (Least Significant Bit) o Bit menys significatiu que consisteix en substituir l'últim bit de cada byte.

  Imaginem que disposem d'una imatge a la que li volem aplicar aquesta tècnica. Cada píxel pot estar representat per 3 bytes corresponents a cada component de color RGB (vermell, verd i blau).

Si es modifica l'últim bit de cada byte, per cada píxel podrem guardar 3 bits d'informació secreta. En el cas de les imatges, modificar l'últim bit de cada byte de dades no modifica la imatge original, o al menys el canvi no és prou substancial com per ser apreciat per la vista humana. De manera anàloga es pot explicar pels arxius de so.

- La tècnica d'addició consisteix en afegir el missatge directament dins el *continent*. És fàcilment detectable i produeix uns canvis de tamany als arxius tan poc comuns que els fa sospitosos.
- La tècnica de generació de nous fitxers es basa en la creació de nous arxius a partir del missatge que es vol ocultar. Un bon exemple d'aquesta tècnica és l'aplicació *Spam Mimic*.

Spam Mimic

Spam Mimic és una aplicació web que produeix missatges de correu del tipus Spam de tal manera que embolcallen d'una determinada manera el missatge ocult. L'e-mail passa desapercebut precisament per ser un missatge *brossa* com tants n'hi ha a Internet.

Tècniques esteganogràfiques en àudio

El present document fa referència a un projecte d'esteganografia basat en àudio. Per tant, a continuació s'exposen les principals tècniques esteganogràfiques per manipular fitxers d'àudio. Es presenten amb el seu nom original en anglès.

Low-bit encoding (Codificació del bit menor)

Aquesta tècnica és l'anàloga a l'LSB d'imatges. Es modifica l'últim bit de cada mostra d'àudio. El principal inconvenient d'aquesta tècnica és que una simple recompressió de format, per exemple de format d'àudio pur al conegut MP3, destrueix la informació oculta que transportava. Un altre inconvenient és que pot arribar a produir soroll detectable.

Spread Spectrum (Eixamplament d'espectre)

Consisteix en generar un senyal d'àudio que pugui ser interpretant com soroll blanc (el típic soroll que sentim a la ràdio quan no aconseguim sintonitzar cap emissora) i posar-hi el missatge ocult dins i afegir el soroll al senyal original.

El problema que presenta aquesta tècnica és que els canals més freqüents per on passa el so, per exemple, els reproductors d'MP3 actuals, incorporen filtres precisament per eliminar el soroll blanc.

Perceptual Masking (Emascarament)

Consisteix en aprofitar que la oïda humana no és capaç de diferenciar dos sons iguals però de diferent nivell. És el mateix esdeveniment que es produeix quan necessitem baixar la veu per parlar confidentment amb una persona que tenim al costat, tot esperant que la resta de converses ambientals emmascararan la nostra.

Echo Data Hiding (Ocultació de l'informació a l'eco)

Aquesta tècnica és l'utilitzada pel projecte *edh* presentat en aquest document. Es tracta de generar un eco del senyal original i amagar-hi la informació. Posteriorment es sumen ambdues senyals i el resultat obtingut és l'arxiu original amb el missatge ocult de manera inapreciable.

Això és pot aconseguir perquè aquesta tècnica explota, novament, una característica de l'oïda humana. Dos sons molt propers en el temps son difícilment detectables.

2. Especificació del problema

Com ja s'ha comentat, el programa *edh* oculta i descobreix missatges de text dins arxius d'àudio.

A continuació s'exposaran els principals punts que cal tenir presents per entendre tota la problemàtica que es presenta en abordar el present projecte.

Tècnica escollida: Eco Data Hiding

La primera qüestió que cal abordar és quina tècnica esteganogràfica d'àudio cal implementar. Anteriorment s'ha presentat les més utilitzades amb els seus avantatges i inconvenients. A priori s'ha de descartar la tècnica de *Low-bit Encoding* per ser la menys robusta. Ens resten:

- ✓ Spread Spectrum
- ✓ Perceptual Masking
- ✓ Echo Data Hiding

De les tres, la menys robusta és l'Spread-Spectrum, doncs ja s'ha comentat anteriorment que un filtratge pot afectar la integritat del missatge ocult.

De les dues he descartat el mètode Perceptual Masking perquè sovint també és considerant com soroll i per tant és susceptible de ser filtrat.

La tècnica escollida és, doncs, l'Echo Data Hiding.

Codificació dels arxius d'àudio

Una altra qüestió a tenir present és si cal preparar el programa per a poder manipular arxius d'àudio als que se'ls hi ha aplicat algorismes de compressió. Per tal de poder-ho fer, caldria implementar algun mecanisme similar als coneguts *plug-ins*, però això queda fora de l'abast d'aquest projecte i de totes maneres es podria implementar com una capa superior que embolcallés l'actual.

Cal tenir en compte les principals característiques d'un arxiu d'àudio. Aquestes son:

- ✓ Freqüència de "mostreig": Es correspon amb el número de mostres (valors) que conté un segon de reproducció. Un valor típic és 44.100 Hz
- ✓ Bits per mostra. Cada mostra (valor) esmentat anteriorment té uns valors possibles. La qualitat de l'àudio vindrà donada en part per aquest valor. Com sabem, amb més bits, més valors decimals podem codificar i per tant més definició podem obtenir. Un valor típic és 16 bits/mostra. El programa només accepta el valor 16 per aquest paràmetre per a simplificar els càlculs i la potència requerida de l'ordinador que ho processi.
- ✓ Número de canals: Es refereix a si el senyal és Mono, Stereo, etc... El programa només acceptarà versions Mono per a simplificar el procés

Codificació del text

Donada l'actual globalització en la que estem immersos, hom podria pensar en internacionalitzar l'aplicació adaptant-la als estàndards de codificació de caràcters, com ara UTF-8, ISO-8859-1, etc... Novament això queda fora de l'abast d'aquest projecte.

El correcte funcionament del programa no es veu afectat per tal problemàtica doncs, en el cas de voler codificar caràcters internacionals la tècnica esteganogràfica restaria invariable. Això ens porta a dues conclusions:

- ✓ Per una banda el programa *edh* només codificarà els primers 127 caràcters de l'estàndard ASCII.
- ✓ Aquesta reflexió ha fet que ens adonem que en futures versions del programa podríem codificar quelcom més que missatges de textos ja que, novament, la tècnica esteganogràfica no ens ho impedeix.

Ocultació de missatges

Entrem ara en el procés d'ocultació de missatges.

Primer cal tenir en compte que la ocultació d'un missatge es fa bit a bit. Per tant, sota el supòsit de codificació de caràcters en ASCII, per cada caràcter del missatge haurem d'ocultar 8 bits.

Com que la unitat mínima d'ocultació és el bit, d'ara en endavant se suposarà que el missatge que es vol ocultar és tan sols la lletra "s". En codificació ASCII el seu valor decimal és el 115. Per tant, volem ocultar la tira de bits següent:

0 1 1 1 0 0 1 1

Partirem d'un senyal d'àudio qualsevol, la representació gràfica del qual podria ser:

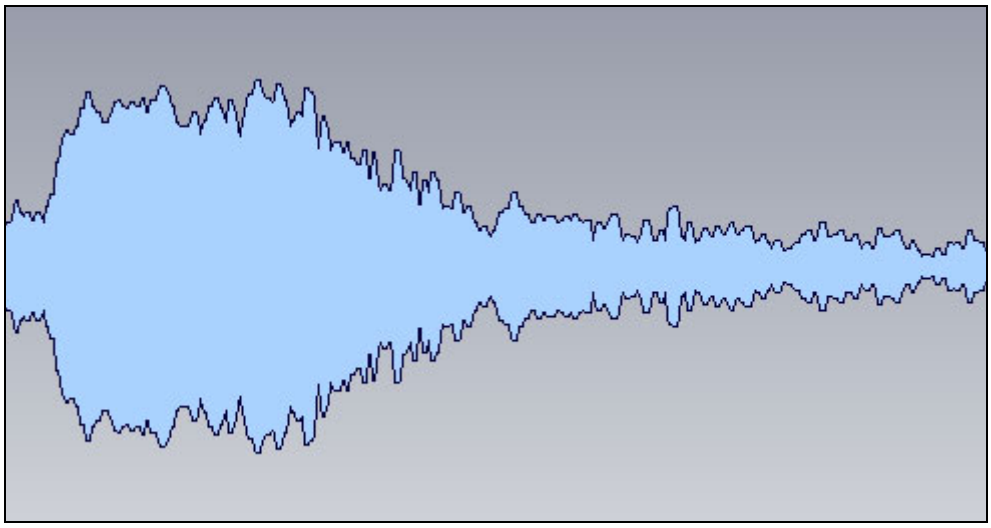


Figura 2

Aquest senyal d'exemple servirà de guia durant tot el procés.

Té una durada de 13s i una freqüència de "mostreig" de 22050 Hz (com ja s'ha comentat anteriorment, els arxius d'àudio acostumen a tenir una freqüència de 44100 Hz). Amb aquesta durada i aquesta freqüència l'arxiu guarda un total aproximat de 286.650 valors.

El procés començarà generant dos senyals amb eco però amb diferent desfasament temporal. Cadascun dels senyals amagarà respectivament tots els ‘1’ i ‘0’ que hi hagi al missatge. Aquest desfasament servirà posteriorment per a que el sistema pugui decidir si en els desplaçaments indicats hi ha un nivell de senyal corresponent a un ‘1’ o a un ‘0’.

Esquemàticament, el resultat de la combinació dels ecos serà la que mostra la següent figura:

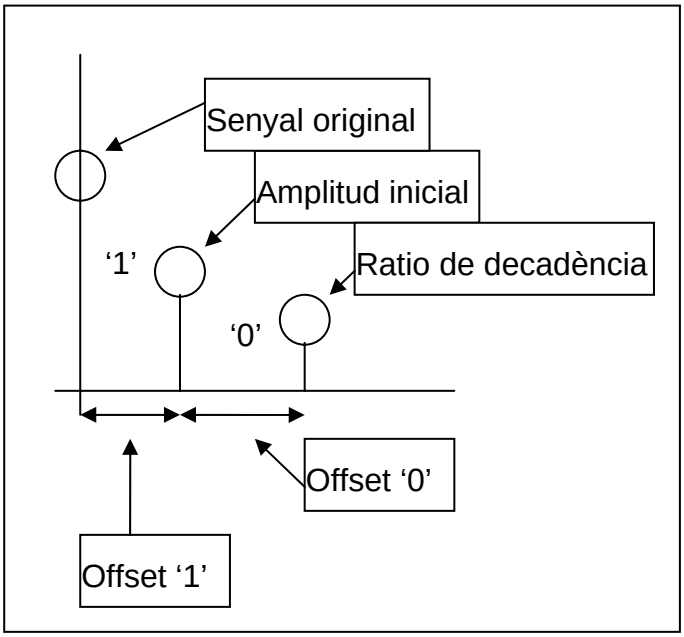


Figura 3

Els paràmetres que s’observen tenen el següent significat:

- ✓ Amplitud inicial: Quan es generin els senyals d’eco, el corresponent a la *tira* d’uns tindrà una amplitud (volum) corresponent a aquest valor.
- ✓ Ratio de decadència: Correspon a quan menor serà el senyal corresponent a la *tira* de zeros en relació a la d’uns.
- ✓ Offset 1: Desplaçament temporal corresponent a un ‘1’
- ✓ Offset 0: Desplaçament temporal corresponent a un ‘0’

Els seus valors son establerts pel programa de manera única per a qualsevol senyal d'entrada. Si es poguessin modificar el destinatari del missatge hauria de conèixer també totes aquestes dades per poder invertir el procés. Més endavant es mostrarà com amb una sola dada ja es podrà descobrir el missatge ocult.

El procés ha creat doncs, dos senyals que contenen el senyal original més l'eco amb els seus respectius desplaçaments i amplitud relativa. El resultat es pot apreciar visualment en les següent figures:

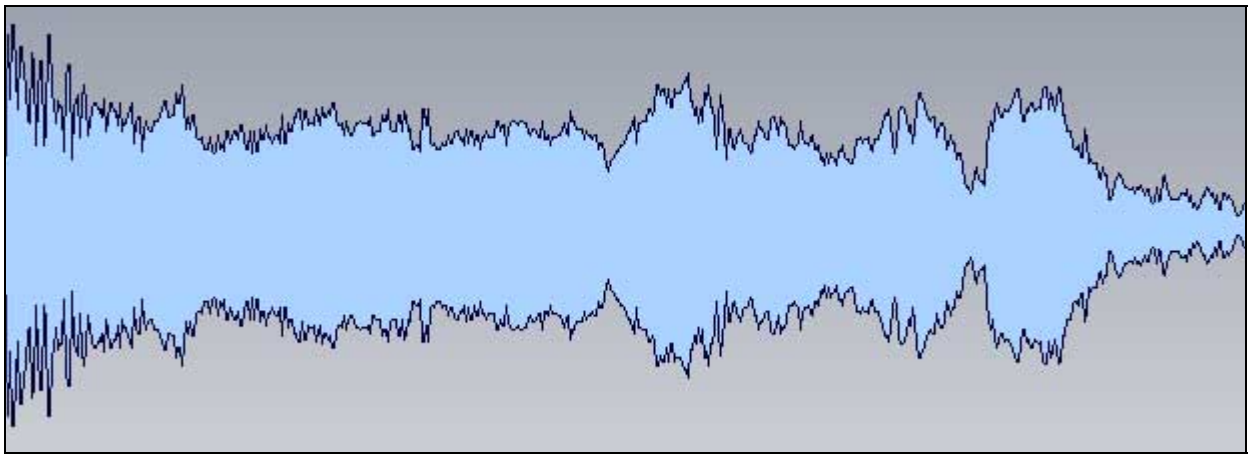


Figura 4. Senyal original més l'eco amb desplaçament Offset 0 = 1.3 milisegons

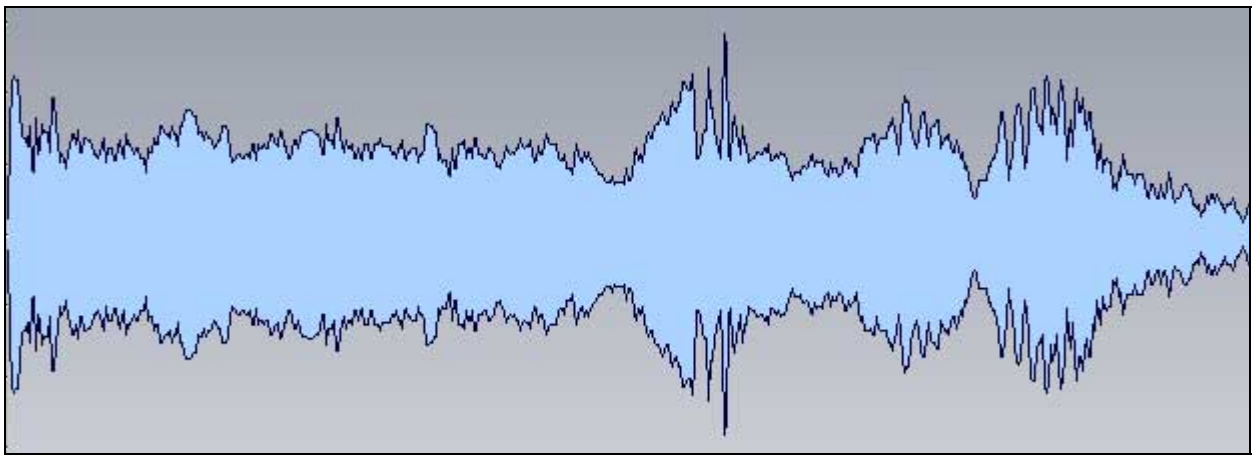


Figura 5. Senyal original més l'eco amb desplaçament Offset 1 = 9 milisegons

Podem apreciar certes diferencies en particular però en general la forma d'ona es manté. Aquests valors de desplaçament fan que l'eco sigui pràcticament inapreciable. Una oïda entrenada pot observar tan sols un cert efecte de reverberació.

El següent pas és calcular el nombre mostres que ocuparà cada bit que volem codificar.

S'ha comentat anteriorment que treballarem només amb un sol caràcter "s", per tant volem amagar 8 bits. Anteriorment, també, s'ha fet referència al nombre de mostres que totals, 286.650. Per tant, dividirem el senyal en 8 blocs de a cada bit, li correspon 35.831 mostres.

Per tal de codificar correctament el missatge a ocultar es creen dos senyals auxiliars que representaran el missatge en codificació binària. Les figures següents aclareixen el concepte:

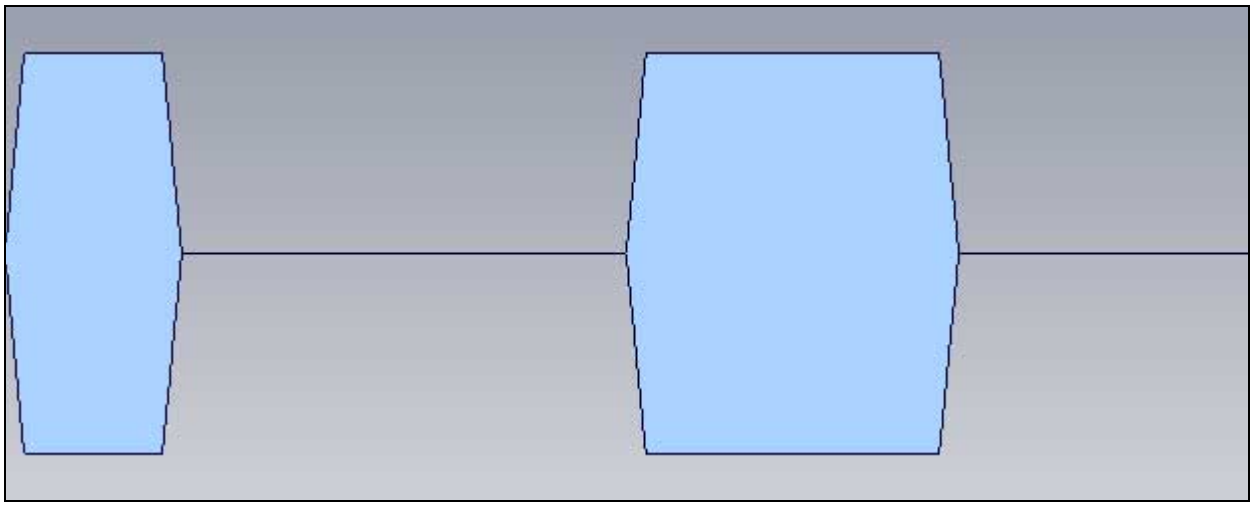


Figura 6. Senyal *Auxiliar 0*

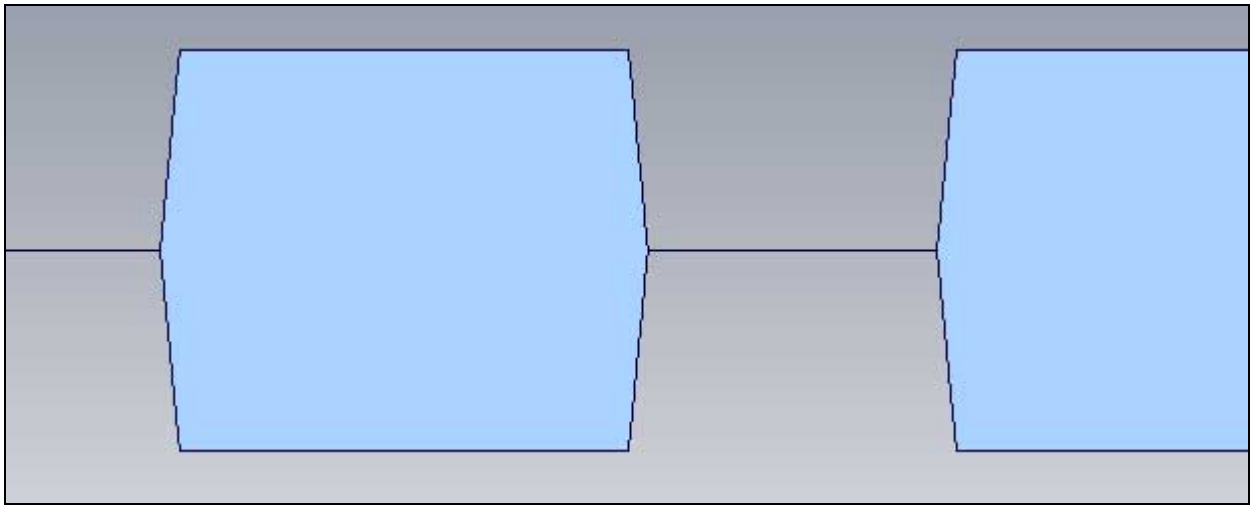


Figura 7. Senyal *Auxiliar 1*

Es fàcil imaginar-se que la figura 7 corresponent al senyal *Auxiliar 1* correspon a un senyal de la mateixa durada que les anteriors dividida en 8 blocs i que cada bloc té valor o no en funció de si el bit corresponen del missatge és 0 o 1. Les següents figures aclareixen aquest últim concepte:

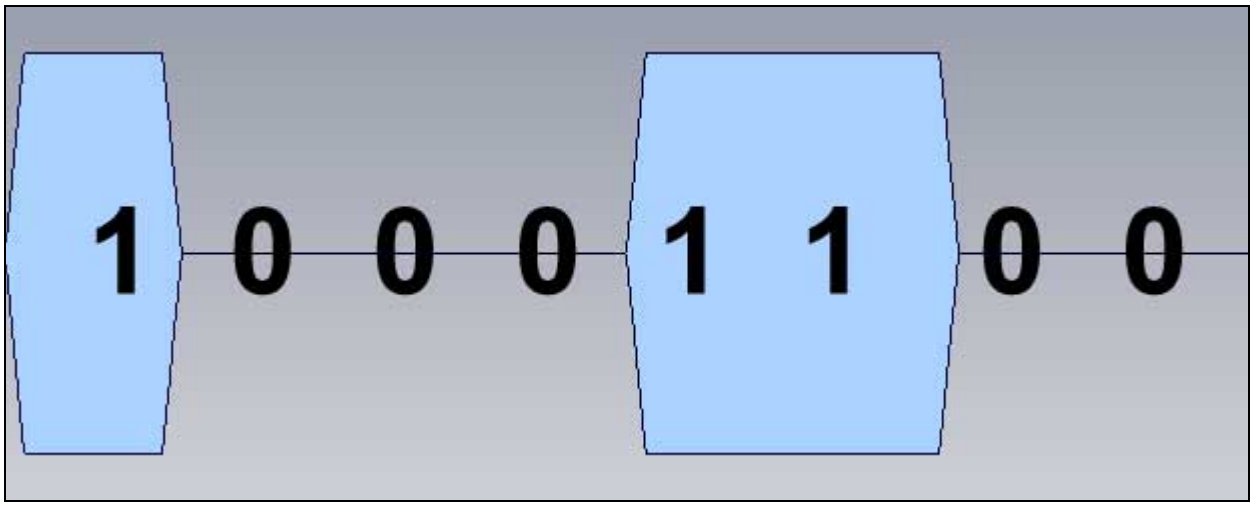


Figura 8. Senyal *Auxiliar 0*

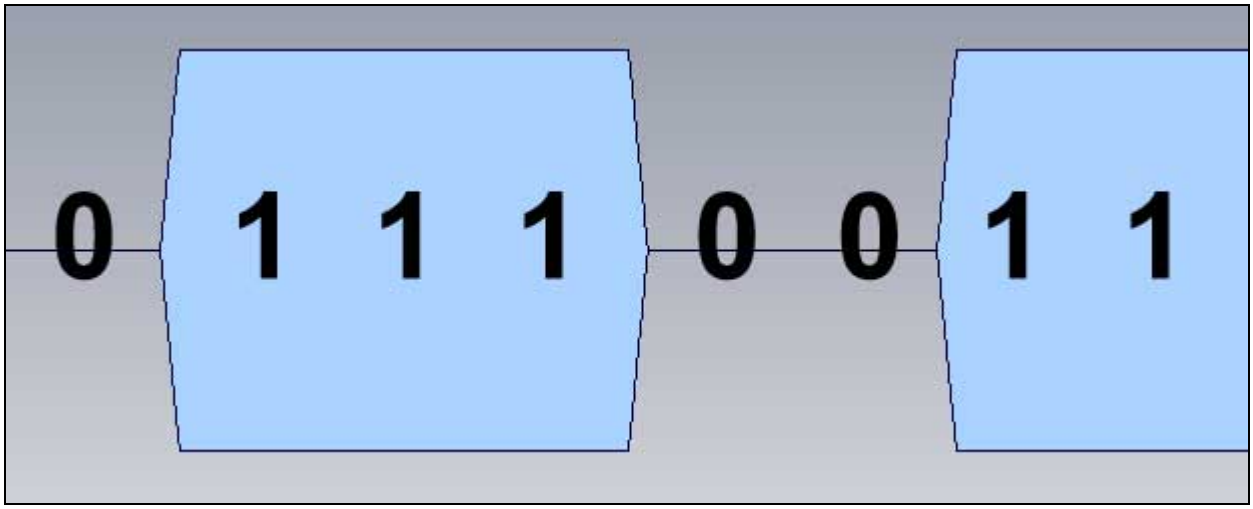


Figura 9. Senyal *Auxiliar 1*

Compari's aquestes figures amb les figures 6 i 7 anteriors per a una millor comprensió. El senyal *Auxiliar* 1 correspon a la representació del missatge a codificar i el senyal *Auxiliar 0* correspon a la representació de missatge negat bit a bit.

Aquestes senyals compleixen una missió. El senyal *Auxiliar 1* es multiplicarà amb el senyal vist a la Figura 4 per tal de tenir un senyal amb eco amb desplaçament *Offset1* allà on el missatge té un bit amb valor 1. Anàlogament es segueix el mateix raonament pel senyal *Auxiliar 0*.

Un cop multiplicades les senyals obtindrem el següent resultat:

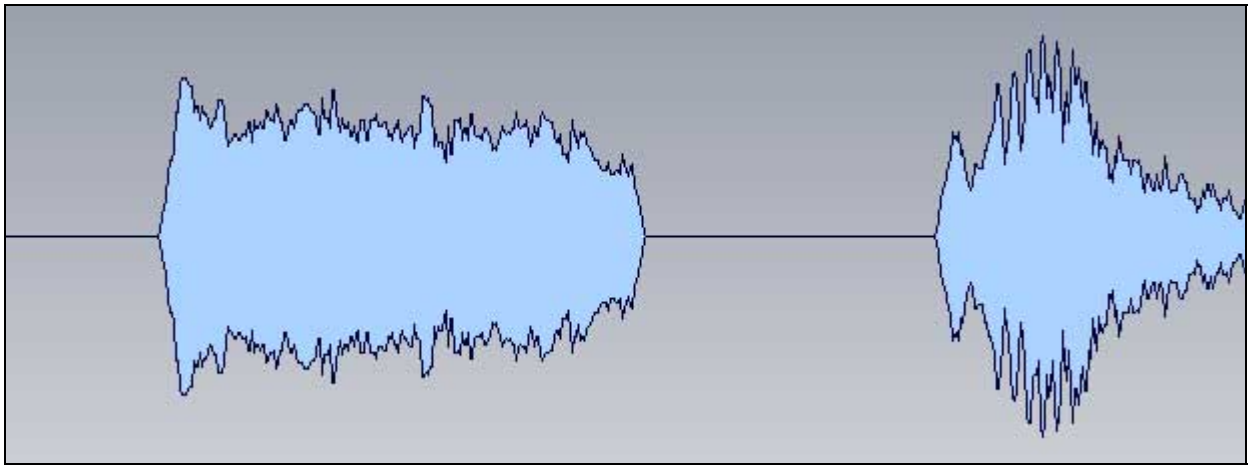


Figura 10. *Auxiliar 1 * Senyal amb eco amb offset1*

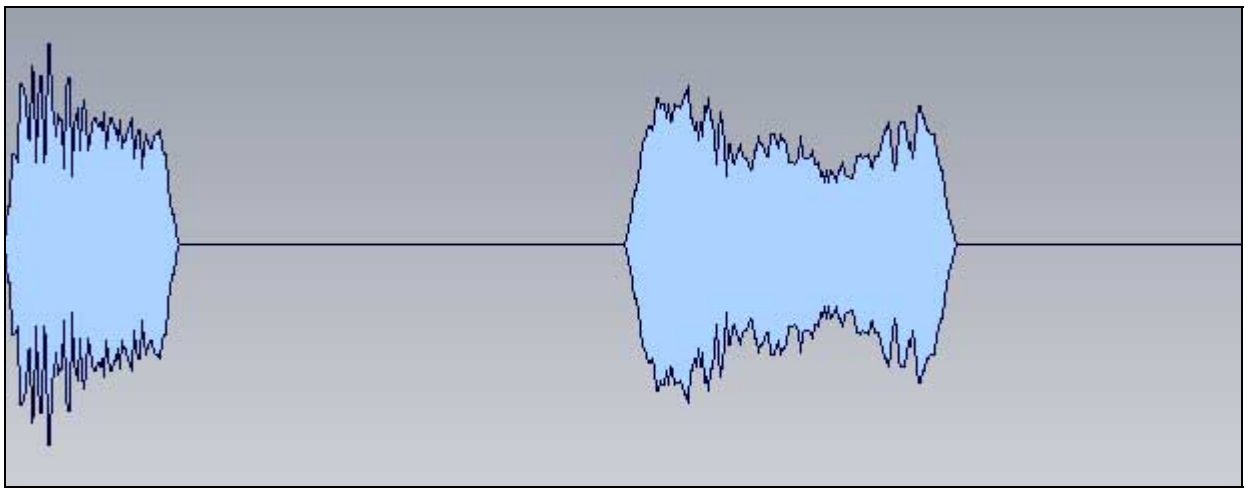


Figura 11. *Auxiliar 0 * Senyal amb eco amb offset0*

En aquest punt tenim porcions de senyal amb un eco inapreciable de diferent desplaçament segons els missatge a ocultar tingui el valor binari ‘1’ o ‘0’.

Ara només manca sumar ambdues senyals resultants. El resultat es presenta a continuació.

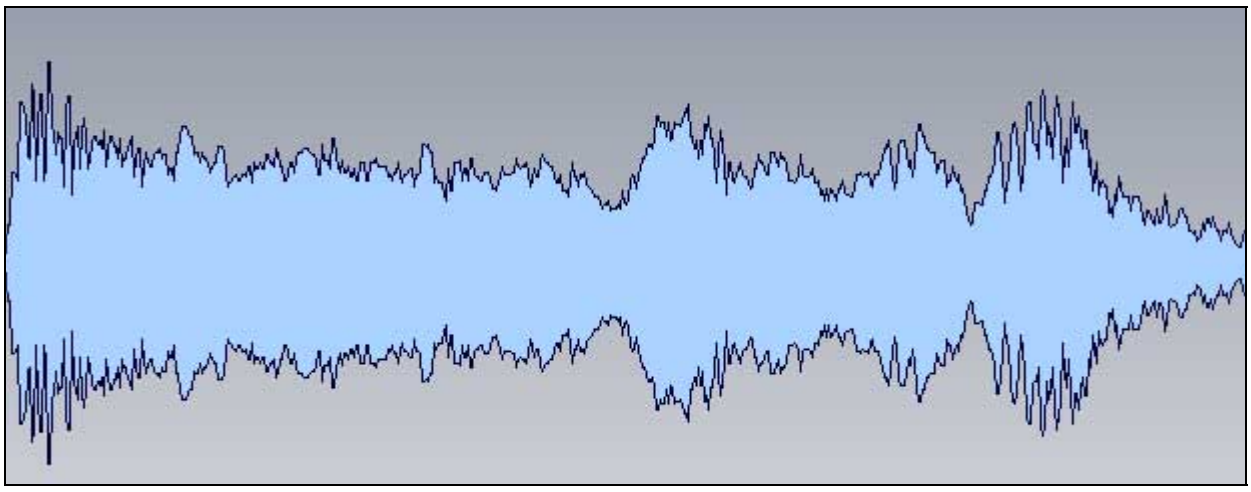


Figura 12. Senyal resultat

Finalment hem obtingut el resultat desitjat. Cal notar dues coses:

- ✓ Si es torna a prendre atenció en les Figures 6 a 9 corresponents als senyals auxiliars es notarà que els canvis de “no senyal” a “senyal” no son abruptes sinó que ho fan a través d’una rampa. Això té una explicació. Si ho féssim de manera abrupta els senyals representats a les figures 10 i 11 també ho serien i per tant, en sumar-les obtindríem 8 blocs la separació dels quals seria discontinua. Això provoca un efecte anomenat “clic” que és totalment perceptible. Cal recordar un cop més que la tècnica d’esteganografia emprada es basa en el fet que passa desapercebuda.
- ✓ Es pot avançar ja que per a invertir el procés el programa li és molt útil saber quina mida té cada bloc, és a dir, quan ocupa cada bit. Si el senyal resultant es divideix en aquests blocs mitjançant l’ús de tècniques d’anàlisi freqüencial es podrà conèixer quin tipus d’eco hi havia en aquella posició i per tant decidir sobre el bit. Així doncs, el programa facilita el número de mostres per bloc per a la seva posterior inversió. Aquest número actua de mode similar a una clau i per això se li ha donat aquest nom.

Descobriment del missatge ocult

Arribats a aquest punt, l'usuari del programa té una clau i un arxiu de so. La millor eina de seguretat, el sentit comú (si es permet la llicència), aconsellaria l'usuari de fer arribar l'arxiu i la clau al destinatari del missatge per diferents canals. D'aquesta manera la sospita que s'estigui transmetent informació delicada minva.

Un cop el destinatari ha rebut els dos components, inicia el procés.

El programa divideix l'arxiu de so en blocs de tantes mostres com especifiqui la clau.

A cada bloc se li aplica un procés que consisteix en calcular el que s'anomena *cepstrum*.

El *cepstrum* d'un senyal és el mateix que calcular el logaritme complex de la Transformada Discreta de Fourier del senyal, obtenint finalment la Transformada Discreta de Fourier inversa. En teoria de senyals es demostra que aquest càlcul és molt útil per descobrir periodicitats i determinades components del senyal com són els *ecos*. Malauradament el *cepstrum* en si, en els senyals d'aquest problema introdueix certa periodicitat que pot confondre el programa a l'hora de detectar '0' i '1's. Aquest problema es soluciona calculant l'autocorrelació del *cepstrum*. Aquesta, bàsicament, informa de com s'assembla un senyal a sí mateixa un cop desplaçada en el temps.

Un cop calculada l'autocorrelació del *cepstrum* per a cada bloc, es procedeix a detectar si en la posició *offset0* hi ha més valor de senyal que en la posició *offset1* per tal de decidir si es tracta d'un '0' o un '1'.

La següent figura mostra la detecció d'un '1':

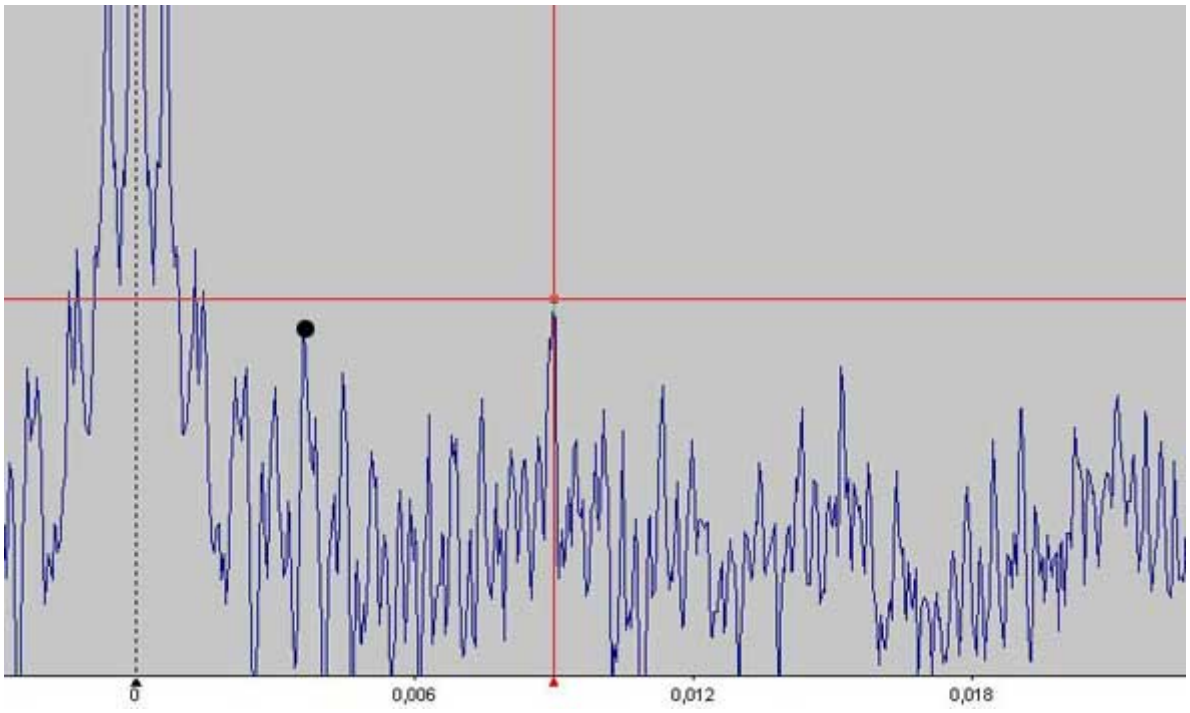


Figura 12. Exemple d'autocorrelació del *cepstrum*

La creu de color vermell indica la posició i valor del senyal a l'instat *offset1.* El valor de l'*offset0*, assenyalat amb un punt negre és inferior.

Cal destacar el següent:

- ✓ La importància de la diferència del V*alor Inicial* i el *Ratio de decadència* és vital per la detecció correcta dels bits.
- ✓ Diferents proves amb diferents fonts d'àudio demostren que certs valors son bons per a certes fonts d'àudio i que cal arribar a un compromís entre elles o bé modificar el programa per a que aquests paràmetres siguin configurables.

3. Disseny del programa i característiques de la implementació

Classes

El programa està escrit en un llenguatge de programació orientat a objectes i per tant està estructurat en classes. Son les següents:

- ✓ Programa: L'execució del programa sempre s'inicia a través del mètode *main* de la classe Programa. La missió d'aquest mètode és decidir quin tipus d'operació es vol realitzar i conduir el fil d'execució cap a un dels dos mètodes principals dels que disposa aquesta classe: amagar i descobrir.

- ✓ SenyalFloat: La classe SenyalFloat és una representació decimal dels valors de les mostres dels senyals d'àudio amb els que treballa l'aplicació. Manté una referència a un objecte de la classe FFTFloat que li servirà per crear la seva Transformada Discreta de Fourier i Transformada Discreta de Fourier Inversa.
  També incorpora mètodes estàtics per a la convolució, suma i multiplicació dels senyals. Més endavant s'explica com s'ha fet.

- ✓ FFTFloat: Classe que manipula dos vectors de dades decimals corresponents a les components real i imaginàries de les Transformades Discreta de Fourier.
  Manté una referència a la seva SenyalFloat per tal de poder actualitzar el valor del senyal en aplicar la Transformada Inversa de Fourier.

- ✓ WaveIn

- ✓ WaveNative

- ✓ WaveStream
- ✓ WaveUtility

Totes les classes que comencen per Wave conformen una llibreria que manipula fitxers d'àudio en format WAV pur.

Mètode amagar

Quan l'usuari indica la operació amagar el fil d'execució passa al mètode *amagar* de la classe *Programa*.

Els passos que se segueixen posteriorment son els esmentats a l'apartat *Descripció del problema*.

En primer lloc amb l'ajuda de les classes *Wave* es llegeix l'arxiu d'entrada i les mostres d'àudio es converteixen a un vector de dades decimals amb les que treballar.

Tot seguit es creen tants objectes tipus SenyalFloat com senyals intervenen en el procés i es van calculant els seus valors de manera seqüencial. En aquest procés hi intervenen mètodes com *Convolucio*, *Suma* i *Multiplicar i crearMixer*.

Mètode descobrir

Una vegada l'usuari proporciona l'arxiu de so i la clau el programa procedeix a la seva apertura i conversió en objecte de tipus SenyalFloat.

Aquest procediment divideix el senyal en tants objectes de tipus SenyalFloat com blocs hagi calculat.

Es recorda que:

- ✓ *Es pot calcular el nombre de blocs ja que el nombre de mostres de l'arxiu d'àudio és conegut i l'usuari proporciona el nombre de mostres de cada bloc mitjançant la clau.*
- ✓ *El nombre de blocs es correspon amb el nombre de bits del missatge ocult continguts a l'arxiu d'àudio.*

Per a cada bloc es calcula l'autocorrelació del seu *cepstrum* amb l'ajuda del mètode *AutocorrelacioCepstrum*.

En aquest punt ja es pot decidir sobre el bit que correspon al bloc actual.

Conforme s'obtenen els bits es van agrupant en bloc de 8 bits per formar els bytes que posteriorment es convertiran a caràcters ASCII.

Mètode crearMixer

Aquest mètode pertany a la classe SenyalFloat i intervé en el procés d'ocultació del missatge.

Donat un missatge, una durada o nombre de mostres totals i una freqüència de mostreig el mètode calcula els senyals que podem observar a les figures 6 a 9. Exactament el mètode calcula els mateixos senyals però amb canvis abruptes dels valors. Un cop el mètode li torna el fil d'execució a la classe que el crida, és responsabilitat d'aquesta suavitzar les discontinuïtats. La manera de fer-ho és aprofitant les propietats de la operació convolució de senyals.

Aquest mètode tan sols ha de recórrer tots els valors del senyal d'entrada i realitzar una operació matemàtica. Per tant el cost computacional es redueix a O(N) on N és el nombre de mostres totals.

Mètode calcularFFT

Aquest mètode pertany a la classe FFTFloat i calcular la Transformada Discreta de Fourier d'un senyal. Utilitiza el mètode FFT perquè és el mètode més senzill que ofereix un cost computacional inferior a d'altres de similar complexitat. El cost és N*O(log N).

Mètode Convolucio

Aquest mètode pertany a la classe SenyalFloat.

És demostrable que la convolució de dues senyals equival a multiplicar les seves transformades i després calcular la transformada inversa. Donat que la operació de multiplicació té un cost O(N) i que el càlcul de la Transformada Discreta de Fourier amb el mètode FFT té un cost N*O(log N), el mètode convolució té aquest cost també.

Suma i Multiplicar

Els mètodes pertanyen a la classe SenyalFloat i tenen un cost computacional d'O(N).

Comentaris i documentació tècnica

Totes les classes pròpies estan completament comentades i auto-documentades. La tecnologia emprada per la construcció del programa ofereix eines per extreure els comentaris i convertir-los a formats de fàcil lectura com ara HTML.

Llibreries externes

Les classes *WaveXXXX* pertanyen a Ianer Munoz i han estat modificades per Corina John. Ambdós autors donen el consentiment per fer un ús privat de les classes.

4. Manual d'instal·lació

L'arquitectura .NET de Microsoft

El programa *edh* està basat en l'arquitectura .NET de Microsoft que es conforma esquemàticament d'una màquina virtual que interpreta un codi intermig que genera el propi programador.

D'aquesta manera, *edh* és un programa compilat a aquest tipus de codi intermig i per tant necessita de l'intèrpret. Aquest intèrpret es pot descarregar gratuïtament de la pàgina Web oficial de Microsoft i el seu nom és *.NET Framework 2.0.*

En el moment d'escriure el present document, l'esmentada arquitectura es pot descarregar de la següent pàgina Web:

Pàgina de descàrrega del .NET Framework 2.0 Redistributable

Un cop descarregat cal procedir a la seva instal·lació seguint els passos que l'instal·lador pugui indicar.

Instal·lació d'Edh

El següent pas consisteix en copiar l'arxiu *edh.exe* a la unitat que es consideri oportú. A partir d'aquí qualsevol llibreria que el programa necessiti la sol·licitarà de manera transparent per l'usuari al *.NET Framework* que s'ha instal·lat en el pas anterior.

A partir d'aquest moment ja es pot començar a utilitzar *edh* normalment.

5. Manual d'ús

El programa treballa en dues vessants: ocultació de missatges (mode *amagar*) i descobriment de missatges (mode *descobriment*). Qualsevol de les dues modalitats funcionen sense intervenció de l'usuari i en mode consola, és a dir, sense interfície gràfica.

En qualsevol dels dos modes es requereixen un mínim de dos arguments.
L'arxiu d'entrada i la modalitat a executar, en aquest ordre.

Mode Amagar (-a)

Aquesta modalitat permet amagar un missatge de text dins un arxiu d'àudio WAV (sense compressió). Per a tal menester *edh* cal especificar la ruta de l'arxiu d'àudio i opcionalment el missatge a codificar. Si aquest no s'especifica el programa amagarà un missatge d'exemple: *"soc un missatge amagat"*. Es pot comprovar que no s'inclouen els accents, doncs *edh,* de moment, només treballa amb caràcters en codificació ASCII.

Aquesta modalitat requereix dos arguments mínims: el nom de l'arxiu d'entrada i el que indica pròpiament la modalitat (el paràmetre –a)

Suposant un arxiu de so anomenat *exemple.wav* i el missatge per defecte, la crida al programa seria la següent:

```
edh exemple.wav –a
```

El resultat de l'execució s'obtindrà en un nou arxiu anomenat *exemple_out_xxxxx.wav*.

Les X corresponen a la clau, un número que el destinatari del missatge hauria d'obtenir per d'altres vies i que necessitarà per invertir el procés.

La resta de paràmetres s'exposen a continuació:

- ✓ *-m* - Missatge a ocultar. Cal que no contingui accents.
  S'escriu -m:"missatge a ocultar"
  Si s'omet s'agafarà per defecte "soc un missatge amagat"
- ✓ *-o* - Ruta de l'arxiu on es vol desar el resultat.
  Si s'omet serà nomArxiu_out_XXXX.wav. S'escriu -o:"sortida.wav"
- ✓ *-D* - Indica que cal guardar/mostrar les senyals/missatges.
  La versió comercial del producte no hauria de contenir aquesta opció.

Mode descobriment (-d)

La segona modalitat en la que treballa *edh* és la de descobriment. Donat un arxiu d'àudio que conté un missatge ocult i una clau numèrica, el programa extreu el missatge que s'hi oculta.

Aquesta modalitat requereix, doncs, tres arguments mínims: el nom de l'arxiu d'entrada, la modalitat (el paràmetre –d) i la clau numèrica.

Suposant un arxiu de so anomenat *exemple.wav* i una clau igual a 1685, la crida al programa seria la següent:

```
edh exemple.wav –d –m:1685
```

El resultat de l'execució s'obtindrà en un nou arxiu anomenat *exemple_out.txt*.

La resta de paràmetres s'exposen a continuació:

- ✓ -o - Nom arxiu sortida. Si s'omet serà nomArxiu.txt.
  S'escriu -o:"nomarxiu_out.txt"
- ✓ -D - Indica que cal mostrar els processos intermitjos a mode de debug.
  La versió comercial del producte no hauria de contenir aquesta

opció.

Solució a problemes

En qualsevol de les dues modalitat el programa pot escriure a la sortida diferents missatges d'error. Es mostren a continuació:

"ERROR: Arguments insuficients."

Aquest error apareix quan no s'especifiquen els arguments mínims per a la modalitat especificada.

"ERROR: Cal especificar el paràmetre m en cas d'utilitzar el mode de descobriment."

Si es vol descobrir el missatge ocult dins d'un arxiu, cal especificar la clau amb el paràmetre *–m*.

"ERROR: Cal indicar quina operació es vol realitzar."

S'ha indicat l'arxiu a tractar però no la modalitat.

"ERROR: No s'ha pogut obrir l'arxiu nomArxiu especificat."

Caldria comprovar que l'arxiu d'entrada i la ruta especificada coincideixen.

"ERROR: L'arxiu nomArxiu no es pot obrir per escriptura. Comprovi que no està obert."

L'arxiu de destí por estar en ús en aquest moment i per tant no s'hi pot escriure. Cal tancar qualsevol procés que en faci un ús exclusiu.

"ERROR: L'arxiu d'entrada no té el format correcte. Comprovi que sigui Mono i quantitzat a 16 bits per mostra."

La versió actual d'*edh* només permet treballar amb arxius d'àudio Mono i quantitzats a 16 bits per mostra. En canvi no hi ha limitació en la freqüència de mostreig, essent valors típics 22050, 44100 i 48000 Hz.

6. Descripció de les proves

Per a provar el correcte funcionament del programa s'han fet diverses proves variant els paràmetres que a priori resulten més crítics. Aquests son:

- ✓ Fonts d'àudio. S'ha provat amb diferents fonts d'àudio de diversa natura per a comprovar que el mètode funciona. Les fonts han estat música instrumental, peces musicals cantades, conversa i totes tenien la mateixa durada.
- ✓ Paràmetres d'amplitud inicial i ratio de decadència. Aquests paràmetres controlen la relació d'amplituds entre els senyals d'eco corresponents als '1' i els senyals d'eco corresponents al '0'. S'ha provat dos conjunts de valors: [1,3 ms, 9 ms] i [1,3 ms, 5 ms]. A priori sembla que el segon conjunt hauria de fallar en la major part dels casos ja que la relació d'amplituds es bastant baixa.
- ✓ Longitud del missatge. S'ha provat amb un missatge curt i un de llarg per provar si això afecta el descobriment del mateix. Els missatges son: "contrasenya", "soc un missatge ocult en un arxiu daudio"

Els resultats més significatius es mostren a continuació en unes taules on s'indica si els resultats han estat satisfactoris (s'ha obtingut el missatge ocult íntegrament) o bé hi ha hagut alguna anomalia (en tal cas es comenta el resultat obtingut).

Prova 1

Missatge curt	[1,3 ms, 9 ms]	[1,3 ms, 5 ms]
Música instrumental	OK	OK
Peça musical cantada	OK	KO
Conversa	OK	KO

Podria semblar que la veu humana interfereix en el procediment. No obstant això, cal notar que la prova només ha fallat en els casos en els que la relació d'amplituds era bastant baixa.

Tot i així, això no vol dir que els resultats que han fallat no continguessin cap caràcter correctament.

En el cas de la Peça musical cantada el missatge tornat ha estat "{ontra33enya"

Si ens fixem veurem que el caràcter "{" té una codificació binària "1111011" i el caràcter correcte "c" té la codificació "1100011". Per tant el programa ha fallat tan sols en dos bits.

Notar també com a curiositat que les dues "s" han estat detectades incorrectament de la mateixa manera. Això pot fer suposar també en un exhaustiu repàs dels algorismes de convolució i càlcul de la Transformada Discreta de Fourier per si son els causants de les transformacions.

Tot i haver assenyalat com a falsa detecció es podria haver considerat com a bona ja que és una paraula suficientment completa com per a interpretar-la.

Prova 2

Conversa	**[1,3 ms, 9 ms]**	**[1,3 ms, 5 ms]**
Missatge curt	OK	KO
Missatge llarg	OK	OK

Novament només ha fallat en els casos en els que la relació d'amplituds és bastant baixa.

Comparant aquests resultats amb els obtinguts en la prova anterior es podria concloure ja no és la veu humana la que interfereix en la detecció sinó que podria haver una relació entre el número de bits codificats i la relació d'amplituds que fes caure la quantitat de deteccions correctes.

Aquest teoria es veu reforçada en la següent prova.

Prova 3

Peça musical cantada	**[1,3 ms, 9 ms]**	**[1,3 ms, 5 ms]**
Missatge curt	OK	KO
Missatge llarg	OK	OK

Novament només ha fallat en els casos en els que la relació d'amplituds és bastant baixa i el missatge és curt. Sembla ser que la relació exposada a la prova anterior existeix. Caldria realitzar proves exhaustives i automatitzades per reafirmar-la.

Conclusions

De totes les proves realitzades, el paràmetre que més ha influït en la detecció correcta dels missatge ocults ha estat la relació d'amplituds de les senyals d'eco. Aquesta era a priori una conclusió prou lògica ja que la resta de paràmetres que es poden combinar tenen valors aleatoris i no premeditats, com per exemple, les fonts d'àudio; son infinites i no segueixen patrons que puguin ser la causa de les falses deteccions.

7. Comentaris i conclusions

L'esteganografia ha estat un art utilitzat esporàdicament al llarg de la història. Una de les últimes aparicions estel·lars va ser el suposat ús que Bin Laden en va fer per cometre els atemptats contra els Estats Units d'Amèrica l'any 2001. Suposadament va fer servir fotografies normals i corrents per amagar-hi tota la informació referent als diferents atacs que s'hi van produir. D'aquesta manera se suposa que a través d'e-mails les hi va fer a arribar a les cèl·lules encarregades d'atemptar.

Tot i que sovint els exemples més destacats corresponen a utilitzacions de tècniques esteganogràfiques en temps de guerra no cal oblidar que l'esteganografia té unes sortides que tot just ara es comença a donar el cas.

Durant l'elaboració del projecte el present redactor va mantenir una xerrada on-line amb l'actual president del MIT Media Lab, el Dr. Walter Bender. En ella el Dr. Bender indicava que ell mateix va formar part del grup de recerca esteganogràfica del MIT Media Lab a finals de la dècada dels 90. Destacava també la importància de l'esteganografia en el món actual i l'empenta que ha sofert aquest art gràcies a Internet. D'altra banda, va comentar que les tècniques de protecció digital de fitxers d'àudio, les DRM, per exemple, no son un bon camp per l'esteganografia degut a que els sistemes esteganogràfics actuals encara son poc robustos.

8. Bibliografia i recursos

Llibres

- ✓ *Codes, ciphers and secret writing (Test Your Code Breaking Skills)*. Martin Gardner. Dover Publications. 1972.
- ✓ *Information hiding techniques for steganography and digital watermarking.*, Stefan Katzenbeisser & Fabien A. P. Petitcolas, Desembre 1999, ISBN 1-58053-035.

Pàgines Web

- ✓ Definició d'esteganografia i vincles a Wikipedia.
  Una bona primera aproximació al concepte i via per obtenir recursos addicionals.
- ✓ Web Kriptopolis.
  Font de diferents articles d'experts en seguretat que ajuden a entendre l'estat d'art.
- ✓ http://www.StegoArchive.com.
  Pàgina web on es poden trobar de manera algunes eines i utilitats esteganogràfiques de pagament.
- ✓ http://www.privacyexposed.com/resources/steganog.htm
- ✓ http://www.jjtc.com/Steganography/
- ✓ Web del programa mp3Stego.
  Programa que utilitza la tècnica de "bit menys significatiu" per amagar la informació.
- ✓ Guillermito Zone.
  Anàlisi de diverses eines esteganogràfiques.

Articles

- ✓ *Echo Hiding,* Daniel Gruhl, Anthony Lu, Walter Bender, MIT Media Lab, 1996.
  El millor document per entendre els conceptes més elementals de l'esteganografia en arxius d'àudio utilitzant la tècnica presentada en el producte referit al present document.
- ✓ *Zollner, Federrath, Klimant, Pfitzmann, Piotraschke, Westfeld, Wicke, Wolf.*
  "Modelling the security of steganographic systems" presentat al Segon Taller d'Ocultació d'Informació, Abril 1998, Portland.
- ✓ *Investigación en Informática. Trabajo Práctico "Esteganografia"*. Universidad John F. Kennedy, 1998
- ✓ *Esteganografia, el poder oculto*. GRRL, Set-Ezine.org
  Breu introducció a la esteganografia per part d'aquest grup d'usuaris.
- ✓ *Steganography, the right way*. Lachlan McGill, SANS Institue, 2005
  Extensa introducció a les tècniques esteganogràfiques en diferents continguts multimèdia.
- ✓ *Steganography FAQ*, Zone-H.Org, 18 Març 2006
- ✓ *MP3Stego. Hiding Text in MP3 Files*, Mark Noto, 15 Setembre 2001
- ✓ *A detailed look at Steganographic Techniques and their usein an Open-Systems Environment,* Bret Dunbar, 18 Gener 2002
- ✓ Mpeg 1.0/2.0 LayersI, II and III header and trailer formats, Laurent Clevi
- ✓ *Detecting Steganographic Content on the Internet,* Neils Provos, 2001
- ✓ *ACM Queue - Cyber warfare steganography vs. steganalysis - For every clever method and tool being developed to hide information*

Exemples, còdi font

Exemples en llenguatge .NET

- ✓ *WaveMixer*
  Exemple de manipulació de fitxers de so en format WAV.
- ✓ *Steganodotnet8*
  Exemple d'aplicació d'esteganografia en fitxers d'àudio utilitzant la tècnica coneguda com LSB (bit menys significatiu).

Altres llenguatges de programació

- ✓ *Llibreria fft*
  Llibreria en llenguatge C que permet l'aplicació de l'algorisme FFT per a calcular la transformada discreta de Fourier d'un senyal.