# criptoStar v1.0

Autor: Vicent Garcia Mesas
Professor responsable: Jordi Herrera Joancomartí
Consultor: Antoni Martinez Balleste
Estudis: Enginyeria Tècnica d'Informàtica de Sistemes
Data: 10 Gener 2006

# ÍNDEX

# 1. INTRODUCCIÓ

## 1.1 ESPECIFICACIÓ DEL PROBLEMA I OBJECTIUS DEL TFC.

Les noves tecnologies han revolucionat la societat actual en la ja anomenada era de la comunicació. La informació viatja a un ritme vertiginós gràcies a les noves tecnologies que permeten, a més a més, emmagatzemar grans quantitats de dades i arxius en dispositius ben petits.

Aquesta facilitat per emmagatzemar i transmetre dades i documents no només té avantatges. El desavantatge principal és la dificultat que suposa mantenir una informació determinada en secret. És a dir, la mateixa facilitat d'emmagatzematge i transmissió fa que, moltes vegades, no sigui gaire segur mantenir un document en un ordinador. D'una banda, la mateixa connexió a internet fa que sigui possible un accés des de l'exterior. D'altra, no sempre el nostre monitor és utilitzat amb exclusivitat i és possible un accés des d'ell mateix però per part d'un altre usuari.

Aquesta necessitat de mantenir les informacions, les dades i els documents particulars en exclusivitat; aquesta necessitat de garantir la privacitat, ha demanat que es desenvolupin sistemes específics de protecció de dades mitjançant el seu xifratge.

El xifratge de documents no només garanteix l'exclusivitat i la privacitat de documents sinó que permet la comunicació xifrada entre dos o més particulars que s'han posat d'acord per a intercanviar-se informació mitjançant una codificació determinada.

Els sistemes de protecció de dades mitjançant el seu xifratge tenen com a objectiu convertir els documents o *textos en clar* en altres documents amb codis inintel·ligibles anomenats *criptogrames.*

Aquest xifratge de documents es portat a terme gràcies a l'aplicació de mètodes diversos: xifres de transposició i substitució, xifres de clau compartida, i xifres de clau pública.

Aquest projecte de final de carrera presenta una aplicació que permet el xifratge de qualsevol tipus d'arxiu mitjançant xifres de clau compartida on la clau de xifratge es obtinguda a partir d'una contrasenya determinada per l'usuari.

Aquesta utilitat, però, ha estat dissenyada de manera que doni resposta als següents objectius de treball:

1. Xifrar i/o desxifrar arxius i carpetes mitjançant l'aplicació de xifres de clau compartida.
2. Dissenyar una aplicació funcional i intuïtiva que sigui de fàcil accés per als usuaris amb coneixements bàsics d'informàtica.
3. Permetre l'accés a usuaris de llengües i orígens diferents.
4. Permetre que l'usuari seleccioni el grau de robustesa o el grau de seguretat depenent del tipus d'algorisme escollit per fer el xifratge.
5. Permetre un sistema de xifratge/desxifratge ràpid accessible des del menú contextual de cada arxiu o un accés des de la interfície principal.

En definitiva, el programa de xifratge que es presenta en el següent treball de final de carrera és una proposta funcional que permet que els usuaris coneixedors de les llengües del disseny ( català, castellà, anglès o francès) xifrin o desxifrin arxius seleccionant un grau de robustesa determinat.

## 1.2 ESTAT DE LA QÜESTIÓ I FONAMENTS

La criptografia[1] és una ciència antiga que està en vigor en l'actualitat. Les noves tendències en criptografia o la criptografia del segle XXI estan vinculades a l'ús de les noves tecnologies i van encaminades a la protecció de la privacitat de les dades transmeses per Internet o, senzillament, emmagatzemades en format electrònic.
En l'actualitat, hi ha diversos sistemes que permeten el xifratge i el desxifratge d'arxius i carpetes. Aquests sistemes es diferencien en el tipus de clau utilitzada.
Una primera classificació segons el tipus de clau[2] ens permetria diferenciar els sistemes de clau compartida o simètrica dels sistemes de clau pública o asimètrica.
Els sistemes de clau compartida es basen en una mateixa clau que s'utilitza per al xifratge i el desxifratge mentre que els sistemes de clau pública assignen dues claus. La primera de les dues claus dels sistemes de clau pública és coneguda per qualsevol usuari mentre que la segona és secreta i només es coneguda per l'usuari propietari de la clau.

L'aplicació de xifratge / desxifratge d'arxius i carpetes que es presenta a continuació utilitza un sistema de clau compartida o simètrica. És a dir, la mateixa clau es utilitzada per al xifratge i desxifratge d'arxius diferenciant-se així del sistema de clau pública que utilitza una clau per a xifrar l'arxiu i una altra de privada per tal de desxifrar els missatges.

El sistema de clau compartida que es presenta, a grans trets, segueix els següents passos:
1.- Obtenir una seqüència pseudoaleatòria a partir de la paraula de pas o contrasenya introduïda i un sistema de xifratge de flux.
2.- Utilitzar la seqüència per a generar una clau secreta el més robusta possible.
3.-Xifrar o desxifrar el contingut de l'arxiu a partir de la clau generada i l'algorisme de xifratge de bloc seleccionat.

1. DOMINGO, J; HERRERA,J; RIFÀ, H (2004:17) "Amb l'aparició dels primers ordinadors, la criptografia deixa de ser un art mil·lenari i esdevé una ciència, l'objectiu bàsic del qual és permetre la comunicació i l'emmagatzematge segur d'informació en presència d'un adversari."

### 1.2.1 Obtenció de la seqüència pseudoaleatòria

El primer pas és l'obtenció d'una seqüència pseudoaleatòria a partir del xifratge simètric de flux tenint en compte que aquesta seqüència ha de complir els tres postulats de Golomb. Aquests postulats són:

1.- Dins el període[1] de la seqüència, la quantitat de zeros i uns ha de ser equivalent o com a màxim diferenciar-se en una unitat.

2.- El nombre total de ràfegues[2] d'una certa longitud dins un període, ha de ser com a mínim $n/2^k$ essent $n$ el nombre total i $k$ la longitud de ràfegues del període.

3.- La funció d'autocorrelació[3] AC(k) només pot prendre dos valors, 1 si la longitud de la ràfega és múltiple de la longitud del període, i un altre valor constant quan no ho sigui.

A banda de complir els tres postulats, la seqüència ha de tenir un elevat grau d'imprevisibilitat de manera que la probabilitat de què el següent bit sigui un 1 o un 0 dins la seqüència estigui propera a 0,5. Per tant, es pot deduir que no serveix qualsevol implementació per a generar la tira de bits. Com a exemples de generadors pseudoaleatòris podríem anomenar el generador de Geffe, el generador de Beth-Piper i el generador Massey-Rueppel.

A la pràctica, a l'hora de generar la seqüència a partir de la contrasenya hi ha dos mètodes diferents descrits tots dos dins les normes *PKCS(public-key cryptography standards)* que són un conjunt d'especificacions desenvolupades en els laboratoris RSA.

Aquestes dos normes són PKCS#5[4] i PKCS#12[5]. Depenent de la norma que s'utilitzi, s'agafaran els buit bits menys pes de cada caràcter (PKCS#5) o bé la totalitat dels bits; setze de cada caràcter amb PKCS#12.

1. DOMINGO, J; HERRERA,J; RIFÀ, H (2004:25) "Període: enter més petit $p$ tal que $s_{i+p} = s_i$ per a tot $i \geq 0$ en què $\{s_i\}_{i \geq 0}$ és una seqüència periòdica".
2. DOMINGO, J; HERRERA,J; RIFÀ, H (2004:26) "Ràfega: conjunt de bits consecutius iguals dins una seqüència".
3. DOMINGO, J; HERRERA,J; RIFÀ, H (2004:25) "Funció d'autocorrelació: nombre de coincidències menys nombre de no-coincidències entre la successió original i la mateixa successió desplaçada $k$ posicions, dividit pel període de la seqüència original".

4: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.pdf

5: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12v1.pdf

### 1.2.2 Generació de la clau derivada

La manera de generar la clau derivada està especificada dins les normes exposades a l'apartat anterior. A banda de la seqüència pseudoaleatòria generada a partir de la contrasenya, és necessari definir uns altres paràmetres per a la generació de la clau secreta. Aquests paràmetres són el *salt* i el *nombre d'iteracions* que farà la funció de derivació.

#### 1.2.2.1 El salt

El *salt* és una seqüència de bits que té com a missió generar diferents claus a partir d'una mateixa contrasenya. Aquest salt està format per 64 bits, si no és conegut, dificulta enormement l'obtenció de la clau secreta per part d'un atacant.

#### 1.2.2.2 El nombre d'iteracions

El *nombre d'iteracions* s'utilitza per a incrementar el cost de producció de claus a partir d'una contrasenya, per tant, també incrementa la dificultat d'obtenir la clau d'una manera il·legítima. Un valor de 1000 iteracions és suficient per a generar una clau prou segura.

#### 1.2.2.3 Funcions de derivació de claus

Una *funció de derivació de claus* produeix una clau derivada a partir d'una clau base i altres paràmetres. La clau base s'obté de la seqüència pseudoaleatòria generada de la contrasenya. D'altra banda, els altres paràmetres són el salt i el nombre d'iteracions definits en els altres apartats.

Els passos que s'han de seguir per a generar la clau són:

1.- Seleccionar el salt i el nombre d'iteracions.

2.- Seleccionar la longitud en octets de la clau.

3.- Aplicar la funció de derivació de claus a la contrasenya, al salt, al nombre d'iteracions i a la longitud de la clau.

4.- Generar la clau derivada.

Es poden diferenciar dos mètodes diferents de funcions de derivació:

1. *PBKDF1(password-based key derivation function 1)* La funció de derivació de la clau basada en contrasenya 1.
2. *PBKDF2(password-based key derivation function 2)* La funció de derivació de la clau basada en contrasenya 2.

#### 1.2.2.3.1 PBKDF1

*PBKDF1* utilitza una funció $hash_1$ , que pot ser MD2, MD5 o SHA1, per a derivar la clau. La seva longitud està limitada a la longitud de la funció *hash* utilitzada. Per a MD2 i MD5, la longitud és de 128 bits i per a SHA1, és de 160. Ateses les limitacions de longitud que comporta la utilització de les funcions *hash*, si és necessari derivar una clau de mida superior als 160 bits, s'ha d'utilitzar l'altra funció de derivació.

#### 1.2.2.3.2 PBKDF2

PBKDF2 utilitza una funció pseudoaleatòria per a derivar la clau. D'aquesta manera la longitud de la clau no està limitada, o si més no, només està limitada per l'estructura de la funció pseudoaleatòria. Un exemple seria la funció HMAC-SHA-1. Aquesta funció obté un *HMAC(message authentication code),* és a dir, un codi d'autentificació d'un missatge basat en la funció *hash SHA-1*.

### 1.2.3 Xifratge/desxifratge de l'arxiu

Una vegada està generada la clau derivada, el proper pas és xifrar l'arxiu mitjançant una sèrie de criptosistemes de xifratge en bloc. Aquests poden ser DES, 3DES i RC2. Depenent del sistema amb què s'ha generat la clau derivada, pot experimentar alguns canvis en el funcionament.

#### 1.2.3.1 Criptosistema DES

És un criptosistema de xifratge de bloc que xifra blocs de dades de 64 bits de llargada mitjançant una clau de 56 bits. Utilitza una sèrie de caixes on es defineixen les permutacions que s'han de fer de les dades. L'algorisme efectua 16 iteracions tot combinant substitucions i transposicions. Tot i que és un criptosistema prou segur, mitjançant maquinari específic és possible realitzar atacs per força bruta per a trencar-lo en pocs dies. Això és possible a causa de la longitud de clau que és insuficient per a les velocitats de càlcul dels ordenadors avui en dia.

1. DOMINGO, J; HERRERA,J; RIFÀ, H (2004:14) "Funció *hash* és una funció que fa correspondre a un contingut de bits de mida variable una representació de mida fixa que sol ser des de 64 fins a 160 bits.

#### 1.2.3.2 Criptosistema 3DES

És un criptosistema de xifratge de bloc que combina el funcionament del DES fent-ho tres vegades seguides. És a dir, xifra el contingut una primera vegada amb una clau. Després, amb una altra , el torna a xifrar i així fins a tres vegades. Això es pot fer de tres maneres diferents:

- DES-EEE3: Tres xifrats DES amb tres claus diferents.
- DES-EDE3: Tres operacions DES amb la seqüència xifrar – desxifrar - xifrar amb tres claus diferents.
- DES-EEE2 y DES-EDE2: Igual que els anteriors però la primera i tercera operació utilitzen la mateixa clau.

Agafant 3DES en front a DES s'aconsegueix doblar el nombre de bits de l'espai de la clau del criptosistema i fer-lo molt més robust i segur als atacs.

#### 1.2.3.3 Criptosistema RC2

És un criptosistema de xifratge de bloc amb la mida de la clau variable dissenyat per Ron Rivest de RSA Data Security. Treballa amb blocs de 64 bits i acostuma a ser entre dues i tres vegades més ràpid que el DES en programari. En funció de la clau seleccionada es pot fer més o menys segur que el DES envers els atacs de força bruta.

#### 1.2.3.4 Xifratge

En el cas que s'utilitzi el sistema PBKDF1 per a generar la clau es pot treballar amb els algorismes DES o RC2 en *mode CBC*[1] . Si s'utilitza el sistema PBKDF2, a banda de DES i RC2 es pot utilitzar també el 3DES.

Els passos a seguir són:

1. Dividir la clau derivada obtinguda en una clau de xifratge amb els 64 primers bits i en un vector d'inici amb els 64 bits restants.
2. Dividir el contingut de l'arxiu a xifrar en blocs adients per al mètode de xifratge. Normalment acostuma a ser de 64 bits per als algorismes exposats. En el cas que l'últim bloc no sigui de 64 bits, s'omplirà amb bits de manera que és pugui saber que aquests bits no formen part de l'arxiu.

1. http://www.uv.es/~sto/cursos/seguridad.java/html "Mode CBC: el mode *CBC(Cipher Block Chaining)* és a dir xifratge en bloc en cadena, és un mode en que es crea un encadenament dels blocs, de manera que el xifratge d'un bloc depèn de l'anterior per mitjà d'un bloc inicial aleatori per al xifratge."

3. Xifrar cada bloc de dades amb sistema de xifratge (DES, 3DES o RC2), mitjançant el mode d'operació (CBC), sota la clau i el vector d'inici obtinguts.
4. Generar una sortida de la mateixa mida que el text en clar, concatenant els blocs xifrats.

#### 1.2.3.5 Desxifratge

Per a obtenir el text en clar d'un de xifrat s'ha de seguir els següents passos:

1. Igual que en el xifratge, s'ha de dividir la clau derivada en una clau de xifratge amb els 64 primers bits i en un vector d'inici amb els 64 restants.
2. Desxifrar el contingut de l'arxiu xifrat amb el sistema de xifratge (DES, 3DES o RC2) mitjançant el mode d'operació (CBC), sota la clau i el vector d'inici obtinguts.
3. Concatenar els blocs resultants del pas anterior per a generar el fitxer .

# 2. L'APLICACIÓ CRIPTOSTAR

L'aplicació ha estat implementada amb el llenguatge de programació *java*, utilitzant la llibreria de classes que porta la versió 1.4.2 de SUN i ampliades amb les llibreries criptogràfiques que porta el proveïdor de seguretat IAIK. Atesa la naturalesa del projecte, s'ha utilitzat un entorn de desenvolupament anomenat *Eclipse* en la seva versió 3.0 per a implementar totes les classes de què consta el projecte.

Dins el directori on s'instal·la l'aplicació, hi ha una sèrie d'arxius i carpetes per al correcte funcionament de l'aplicació a banda de les classes i els arxius executables i accessos directes. Hi ha un arxiu html amb el manual d'usuari de l'aplicació, una carpeta anomenada *lib* on es poden trobar les llibreries criptogràfiques IAIK. D'aquesta manera l'usuari no s'ha de prendre la molèstia d'instal·lar tot el paquet criptogràfic. N'hi ha una altra anomenada *idioma* on s'ubiquen els arxius responsables del canvi d'idioma de l'aplicació. La desaparició o corrupció d'algun arxiu d'aquesta carpeta no és crítica però l'aplicació no podrà funcionar amb l'idioma en concret de què hagin desaparegut arxius donant una sèrie d'errors.

Hi ha també una altra carpeta anomenada *imatges* on es troben totes les imatges i icones que utilitzen les diferents interfícies del programa. N'hi ha una altra anomenada *polítiques_seguretat* on hi ha les polítiques de seguretat sense restriccions de longitud de claus.

A banda, i per a tenir un control més eficient dels arxius xifrats i desxifrats, hi ha dos carpetes per a ubicar-los on per defecte l'aplicació els desarà. També es pot trobar una altra carpeta anomenada *TempFiles* que el programa utilitza per a generar arxius temporals que esborra una vegada s'ha tancat l'aplicació.

Totes aquestes carpetes es visualitzen en la imatge següent:

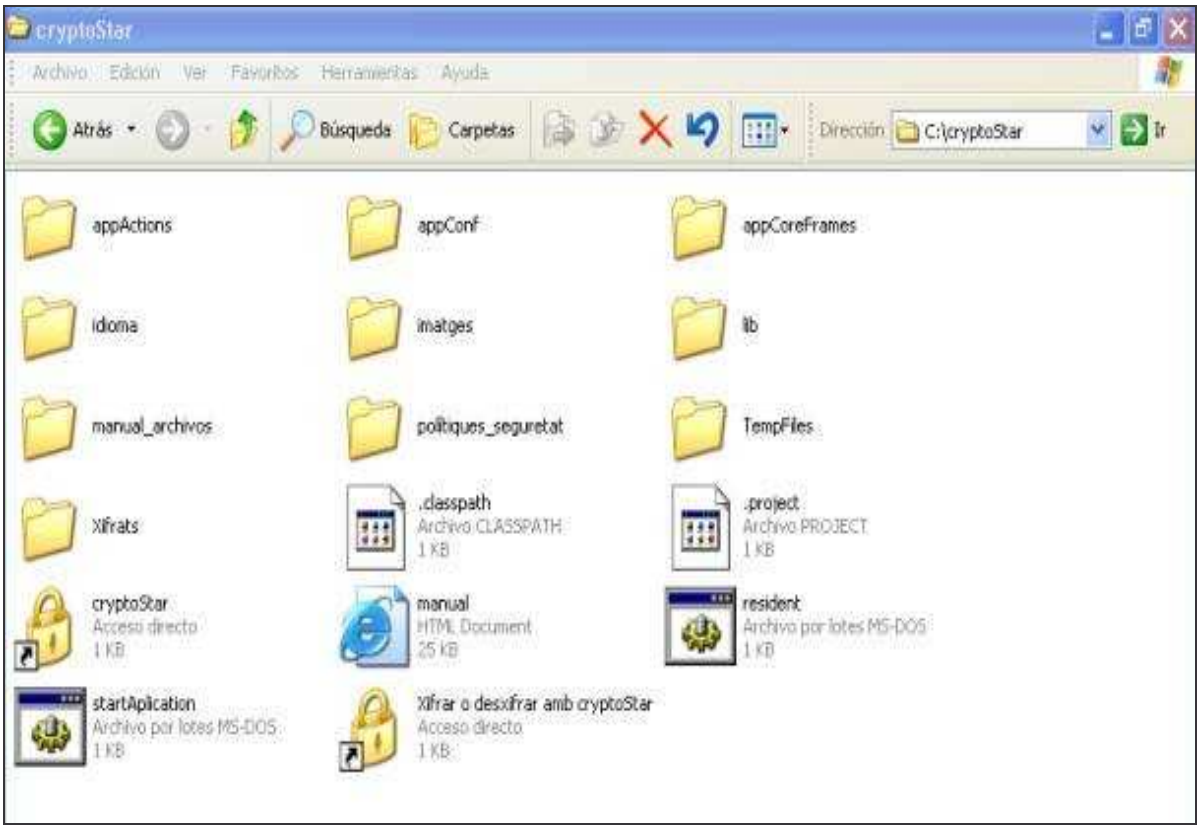



*Figura 1: Directori del programa. FONT: CriptoStar v1.0*

## 2.1 DISSENY DEL PROGRAMA O MÈTODE IMPLEMENTAT

L'aplicació està dividida en paquets definits segons la seva funcionalitat per a una millor ordenació de classes. En cada un, hi ha una sèrie de classes que s'encarreguen d'implementar totes les funcionalitats que pot oferir el programa.

Atès que *java* és una tecnologia de programació orientada a objectes, l'estructuració de les classes ha estat pensada en aquest sentit. Per tant, s'ha procedit a definir cada classe com un objecte amb les seves variables i els seus mètodes. Això ha estat possible amb l' excepció de les dues classes que generen les interfícies gràfiques de l'aplicació on el disseny està més vinculat a la senzillesa i funcionalitat que no pas a les pautes pròpies del procediment de creació orientat a objectes.

Es poden distingir tres paquets diferents:

- *AppCoreFrames*. On s'ubiquen les classes que generen les dues interfícies de què consta l'aplicació: *InterficiePrincipal.class* i *FrameResident.class.*

- *AppConf*. On s'ubica el fitxer de configuració del programa i la classe encarregada de gestionar-lo per a una correcta selecció de l'idioma, proveïdor de seguretat i algorisme de xifratge: *ConfiguracioAppManager.class.*
  A banda també podrem trobar el fitxer de configuració: *config.properties.*
- *AppActions*. On s'ubiquen tota una sèrie de classes per a fer totes les accions i funcionalitats que permet l'aplicació. Dins d'aquest paquet podem trobar les següents classes d'utilitat : *ArxiuConfiguracio.class, EsborraCarpetes.class, IdiomaManager.class, Xifrador.class, ZipManager.class, Manual.class, MonitorExtract.class* i *RetornaParametres.class.*

A banda, també es poden trobar tres classes que no pertanyen a cap paquet atès que són les que s'encarreguen de la instal·lació de l'aplicació al disc. Aquestes classes són *Instalador.class, ZipManager.class i MonitorExtract.class .*

Tot seguit s'especificarà el disseny i les funcionalitats de cada classe per separat per tenir una idea acurada de les funcions que fa cadascuna dins l'aplicació.

### 2.1.1 La classe InterficieGrafica

#### 2.1.1.1 Descripció

Aquesta classe és la porta d'entrada a l'aplicació. S'encarrega de gestionar totes les opcions i paràmetres per a poder xifrar o desxifrar l'arxiu. Implementa un marc per a ubicar el menú, els botons els camps d'introducció de l'arxiu i contrasenya, i les caixes de text per a la visualització del contingut de l'arxiu xifrat i a xifrar. Des d'aquí es pot personalitzar el programa amb un determinat idioma, un algorisme i un proveïdor de seguretat per a xifrar els fitxers. A més, hi ha un accés per a visualitzar el manual d'usuari en format html i una petita descripció del programa.

#### 2.1.1.2 Disseny

Dins la classe, es dissenya l'aspecte que ha de donar la interfície i es gestiona totes les dades que es necessiten per a poder xifrar o desxifrar els arxius. El disseny i funcionalitats que ofereix són les següents:

1. Crear una instància de la classe JFrame per ubicar tots els contenidors de dades.
2. Generar un seguit de components necessaris per a que el programa funcioni correctament. Per exemple, etiquetes per a indicar el mode de treball de

l'aplicació i el tipus d'algorisme a utilitzar. Un botó per a seleccionar l'arxiu o carpeta mitjançant un cercador. Camps de text per a introduir l'arxiu, la contrasenya, i la confirmació per assegurar que allò que s'ha escrit és allò que es volia escriure. Hi ha dos opcions per a permetre que l'arxiu resultant sigui comprimit i que l'arxiu seleccionat s'esborri una vegada acabada l'operació demanada. A més a més, hi ha un botó que indica l'inici de l'acció de xifratge o desxifratge. També hi ha dos capses de text per a visualitzar el contingut dels arxius tant el xifrat com el que hem de xifrar i en el cas que es seleccioni una carpeta, dins la caixa de text ho avisa.

3. Generar un menú amb totes les opcions de que disposa el programa distribuïdes amb submenús. Algunes d'elles són accessibles mitjançant tecles d'accés ràpid per a fer més intuïtiva i pràctica l'aplicació. Aquí tenim les opcions del mode de treball(xifratge o desxifratge), tancar l'aplicació, deixar-la resident, selecció de l'algorisme i proveïdor de seguretat on hi ha fins a cinc opcions diferents, tria de l'idioma de l'aplicació i dels missatge d'informació que pugui donat el programa , i un manual d'ajuda en línia.

Si es prem el botó del cercador per a seleccionar un arxiu o carpeta, depenent del mode en que es trobi l'aplicació, s'ubicarà la cerca inicial en diferents llocs i, fins i tot, introduirà filtres en l'extensió dels arxius a cercar. Per exemple, en mode xifrador, on s'ha de cercar qualsevol arxiu o carpeta no hi ha cap filtre i la cerca comença al directori per defecte del sistema. Quan està en mode desxifrador, la cerca inicial va al directori on resideixen els arxius xifrats per defecte, és a dir, a la carpeta *xifrats* i s'aplica el filtre per a què només mostri els arxius amb extensió crs.

Dins la mateixa classe i per a habilitar la utilització de l'algorisme 3DES, si s'engega l'aplicació es canvien les polítiques de seguretat que porta per defecte la versió de java per unes altres de més permissives. Després es deixen instal·lades una altra vegada les que portava inicialment quan es tanca el programa. Des d'aquí també actualitza el fitxer de configuració amb les opcions que s'agafen per a què estiguin disponibles quan es torni a obrir el programa o es cridi des del menú contextual.

A l'hora de xifrar o desxifrar l'arxiu, no és necessari seleccionar amb el ratolí cada component atès que l'aplicació respon al teclat. Per defecte s'escriu dins al quadre de l'arxiu i quan es prem la tecla *intro,* passa al camp de text de contrasenya. Si es torna a pitjar, passa a la de confirmació etc.

El control d'errades és fa localment a cada classe i atesa la seva naturalesa, és on hi ha més fets a controlar. A grans trets, l'aplicació controla les errades i opcions especials mitjançant missatges d'avís, error i confirmació segons calgui. Si es prem el botó per a xifrar o desxifrar l'arxiu o carpeta i es marca alguna dada, o bé la contrasenya té menys de 6 símbols o bé no coincideix amb la confirmació introduïda o, fins i tot, si s'introdueix la direcció d'un arxiu que no existeix, l'aplicació mostrarà un missatge especificant el motiu de l'error. Tanmateix mostrarà missatge de confirmació quan es desitgi un canvi d'idioma. Es selecciona l'opció d'esborrar l'entrada, l'opció de comprimir dades o sortir de l'aplicació. Si l' operació de xifratge o desxifratge ha fracassat, ho indicarà amb un missatge d'error i si és un èxit, també ho indicarà.

Qualsevol arxiu o carpeta, previ al xifratge, s'empaqueta mitjançant la classe ZipManager. Una opció interessant que s'en deriva és el fet de controlar al desxifratge si s'ha introduït la contrasenya correcta o bé l'arxiu xifrat ha estat manipulat.

Aquest fet té l' avantatge que si s'ha seleccionat l' opció d'esborrar l'arxiu i no s'ha posat la contrasenya correcta, l'arxiu no desapareixerà i a més podem tenir la certesa que l'arxiu no ha esta manipulat i per tant és íntegre i de confiança.

El seu desavantatge és la facilitat d'una cerca exhaustiva de la contrasenya correcta per un usuari il·legítim, cosa que s'atenua si en posem una prou llarga d' almenys 6 símbols.

#### 2.1.1.3 Classes importades

Per al correcte funcionament del programa ha estat necessari importar les següents classes i paquets de les llibreries de java:

***java.awt.**** **i** ***javax.swing.****

Per al disseny de la interfície gràfica i missatges. Aquestes classes ajuden al desenvolupament d'aplicacions gràfiques en java i aporten una sèrie de funcionalitats per a gestionar els diferents fets que poden ocórrer quan es treballa amb el teclat i el ratolí.

***Java.io.File, Java.io.FileInputStream*** **i** ***Java.io.FileOutputStream***

Classes per a treballar amb arxius que aporten funcionalitats per a la gestió de l'escriptura, selecció i lectura de fitxers .

## 2.1.2 La classe FrameResident

### 2.1.2.1 Descripció

Aquesta classe és la responsable de crear una altra interfície més compacta que es cridarà quan el programa treballi resident en memòria i és cridi mitjançant un accés directe al menú contextual. Aquest menú surt quan es selecciona un arxiu o carpeta prement el botó dret del ratolí i l'accés directe es troba ubicat dins el submenú *enviar a ...*


*Figura 2: Menú contextual d'un arxiu.. FONT: CriptoStar v1.0*

Aquesta interfície no té menú, no li fa falta perquè les opcions de configuració s'han de seleccionar des de l'altra interfície. D'altra banda, les opcions d'algorisme i proveïdor els treu o bé de l'arxiu de configuració o bé de l'arxiu xifrat. L'idioma de la interfície també procedeix de l'arxiu de configuració.

### 2.1.2.2 Disseny

Aquesta classe s'ha definit a partir de l'ampliació d'una altra, JFrame, que es troba a les llibreries de java. Dins la classe, es dissenya l'aspecte que ha de donar la interfície i es gestiona totes les dades necessàries per a poder xifrar o desxifrar els arxius.

Atès que es crida a partir del menú contextual d'un arxiu o carpeta, quan s'entra a l'aplicació automàticament es posa el nom de l'arxiu dins el camp de text que li correspon i es posa en

el mode adient a partir de l'extensió de l'arxiu. És a dir, si l'arxiu té extensió crs es posa en mode desxifrar, en qualsevol altre cas, en mode xifrar.

Dins la interfície podem distingir cinc camps de text: tres d'ells ubiquen l'algorisme, el proveïdor i l'arxiu que no estan habilitats per a canviar-se atès que aquestes dades les ha d'obtenir d'una manera transparent a l'usuari. Hi ha dos camps més per a indicar la contrasenya i la confirmació. A banda, també es poden trobar les opcions d'esborrar l'entrada i comprimir les dades que funcionen de la mateixa manera que a la interfície principal.

A la part inferior del marc, podem trobar un seguit de botons:
*Torna a l'aplicació* : Tanca la finestra obrint-se l'aplicació principal amb la diferència que l'arxiu seleccionat és manté.
*Acceptar:* Si tot està correcte, xifra o desxifra .
*Cancel·la* : Tanca la finestra deixant-la resident.
*Tancar aplicació* : Tanca totalment l'aplicació.
Els sistemes de xifratge i desxifratge, control d'errades i missatges d'informació són equivalents a l'altra classe.

#### 2.1.2.3 Classes importades

Aquesta classe utilitza bàsicament les mateixes classes i paquets importats que a la interfície gràfica principal.

### 2.1.3 La classe ConfiguracioAppManager

#### 2.1.3.1 Descripció

Aquesta classe s'ocupa de crear i gestionar un arxiu de propietats que utilitzaran les altres classes per a obtenir algun paràmetre de configuració. Dins l'arxiu es poden trobar la data de l' última modificació de l'arxiu, l'idioma, l'algorisme de xifratge i el proveïdor seleccionat per l'usuari.

#### 2.1.3.2 Disseny

La classe és una extensió de *Properties*, una altra definida dins les llibreries de java que s'utilitza per a definir i gestionar d'una manera eficient arxius de propietats que responen a la sintaxi *clau = valor* on la clau és el text o paraula clau per a ubicar una propietat i valor seria el contingut d'aquesta propietat.

Presenta una sèrie de mètodes per a la inserció i recuperació de les dades de l'arxiu i la càrrega i guarda de les dades a l'arxiu. A banda, en el cas que l'arxiu esdevingui corrupte o s'esborri accidentalment, es torna a crear amb les opcions per defecte. Aquestes opcions són l'idioma català, l'algorisme DES i el proveïdor SUN JCE.

#### 2.1.3.3 Classes importades

Com a classes especials que apareguin aquí podríem anomenar :

***java.util.Properties***

Classe per a gestionar un arxiu de configuració que conté tots els mètodes necessaris per a la càrrega i selecció d'opcions les quals interessa fer persistents una vegada es tanqui l'aplicació.

### 2.1.4 La classe Xifrador

#### 2.1.4.1 Descripció

Aquesta classe és la que aporta l'essència del projecte atès que és l'encarregada del xifratge/desxifratge dels arxius i per tant és aquí on s'utilitzen les classes importades de les llibreries de java i IAIK per a portar a terme tot el sistema de xifratge basat en contrasenya que ja s'ha explicat a la part de fonaments. Com es podrà apreciar java facilita una sèrie de classes per a fer que sistema de xifratge sigui molt senzill d'efectuar, estalviant temps de disseny i línies de codi al programador.

#### 2.1.4.2 Disseny

Aquesta classe, com ja s'ha comentat prèviament, és la que s'encarrega de tot el procés de xifratge i desxifratge de l'arxiu. Ara bé, abans de fer l'operació en si, se li ha de facilitar una sèrie de paràmetres.

En un principi, el sistema de xifratge només permetia utilitzar un mètode i un proveïdor. A mesura que ha anat avançant el projecte se li han afegit una sèrie de mètodes per a agilitar el canvi dinàmic d'algorisme i proveïdor.

Atès que l'aplicació suporta dos proveïdors de seguretat, quan escau, la mateixa classe instal·la a l'arxiu persistent que té java de proveïdors, una instància de IAIK per a què el reconegui com a tal, esborrant-la una vegada la configuració del programa ja no la necessiti.

Si per causes externes a l'aplicació no es pogués carregar amb un algorisme o proveïdor registrat, el xifrador ho indicaria i posaria les opcions que té per defecte, per a habilitar un xifratge de l'arxiu.

Aquestes causes poden esdevenir a causa d'una manipulació de l'arxiu de configuració o una errada interna del sistema i, per tant, no han de ser freqüents.

Quan es crea una instància del xifrador es defineix el salt que tindrà un valor constant i igual per a tots els arxius que es xifrin del programa. El nombre d'iteracions, tal i com aconsella les normes PKCS#5, serà de 1000. Al mateix temps, s'ompliran les variables de l'algorisme i proveïdor amb els valors per defecte.

Els algorismes que disposa l'aplicació són:

- DES SunJCE: *PBEWithMD5AndDES*
- DES IAIK: *PBEWithMD5AndDES_CBC*
- 3DES SunJCE: *PBEWithMD5AndtripleDES*
- 3DES IAIK: *PbeWithSHAAnd3_KeyTripleDES_CBC*
- RC2 IAIK: *PbeWithSHAAnd40bitRC2_CBC*

*PBEWithMD5AndDES*

Aquest algorisme utilitza una funció *hash* MD5 per a crear la clau i xifra/desxifra l'arxiu amb el criptosistema DES

*PBEWithMD5AndDES_CBC*

Aquest algorisme utilitza una funció *hash* MD5 per a crear la clau i xifra/desxifra l'arxiu amb el criptosistema DES, utilitzant el proveïdor de seguretat IAIK.

*PBEWithMD5AndtripleDES*

Aquest algorisme utilitza una funció *hash* MD5 per a crear la clau i xifra/desxifra l'arxiu amb el criptosistema 3DES.

*PbeWithSHAAnd3_KeyTripleDES_CBC*
Aquest algorisme utilitza una funció *hash* SHA-1 per a crear la clau i xifra/desxifra l'arxiu amb el criptosistema 3DES, utilitzant el proveïdor IAIK.

*PbeWithSHAAnd40bitRC2_CBC*
Aquest algorisme utilitza una funció *hash* SHA-1 per a crear la clau i xifra/desxifra l'arxiu amb el criptosistema RC2, utilitzant el proveïdor IAIK.

#### 2.1.4.3 Classes importades

Com a classes especials utilitzades podríem anomenar :

***javax.crypto.spec.PBEKeySpect***
La funció d'aquesta classe és la de generar una especificació de clau criptogràfica a partir dels bits que conté la contrasenya. Hi ha diferents mecanismes per a generar la clau i aquests es reflexen dins els estàndards PKCS #5 , on només s'agafen els 8 bits de menys pes de cada caràcter i PKCS #12 , on s'agafen els 16 bits de cada caràcter per a generar la clau.

***javax.crypto.spec.PBEParamSpect***
Aquesta classe s'encarrega de proporcionar els paràmetres que haurà d'utilitzar el xifrador i per tant s'haurà de crear una instància que contingui el valor del salt i en nombre d'iteracions que volem.

***javax.crypto.spec.SecretKeyFactory***
Aquesta classe representa un magatzem de claus criptogràfiques. La funció d'aquesta classe és la de proporcionar claus per a xifrar arxius a partir d'especificacions de claus.
Aquesta classe és bidireccional, és a dir, serveix tant per a generar claus a partir d'especificacions com per a generar especificacions a partir de claus. Per al primer cas, s'utilitza el mètode *generateSecret* i per al segon, s'utilitza getKeySpec.

***javax.crypto.SecretKey***
Aquesta classe és la responsable de crear una clau criptogràfica a partir de les especificacions extretes de les classes anteriors. Aquesta clau ja es pot utilitzar per a xifrar i desxifrar fitxers.

***javax.crypto.Cipher***

Aquesta classe implementa el funcionament del xifrador per a què treballi funcionalment segons les característiques que li enviïs. En el nostre cas s'ha de preparar un xifrador basat en contrasenya i això s'aconsegueix enviant-li com a paràmetres les instàncies creades dels paràmetres amb la classe *PBEParamSpect.*

***javax.crypto.CipherOutputStream i javax.crypto.CipherInputStream***

Aquesta classe proporciona un mètode d'emmagatzematge dels arxius xifrats o desxifrats per a una posterior lectura o escriptura al disc.

***iaik.security.provider.IAIK*** Aquesta classe proporciona una instància del proveïdor de seguretat IAIK que conté totes les dades necessàries per a fer-lo servir.

***java.security.Security***

Aquesta classe proporciona mètodes per a inserir i eliminar proveïdors dins l'arxiu de seguretat que té persistent la màquina virtual de *java.*

## 2.1.5 La classe ZipManager

### 2.1.5.1 Descripció

Aquesta classe s'encarrega de gestionar tot el vinculat amb la compressió i descompressió d'arxius i carpetes mitjançant les utilitats de què java disposa. En un principi, només les carpetes es comprimien. Tanmateix, atès que mitjançant l'empaquetament de qualsevol arxiu abans del xifratge es pot saber si el desxifratge ha estat correcte, s' ha optat per fer passar a tots els arxius.

Quan es selecciona una carpeta per a xifrar, tenint en compte que pot contenir múltiples arxius, una manera eficient i ràpida de fer-ho és comprimir-la prèviament i convertir la carpeta en un sol arxiu. Tot i amb això no és obligat que el contingut de l'arxiu resultant estigui comprimit. De fet, a l'aplicació hi ha una opció per a què l'usuari decideixi si ho vol fer o no.

No obstant, tenint en compte la riquesa de caràcters en les llengües llatines, la compressió d'arxius amb les classes que proporciona *java* no es comporten degudament. Es pot

comprovar a l'adreça que té SUN[1] a la web que existeix un error de programació amb aquestes classes de manera que no actuen bé amb caràcters accentuats.
Això no suposa cap problema sempre i quan comprimim i descomprimim amb les classes que proporciona *java*.

#### 2.1.5.2 Disseny

La funció d'aquesta classe està ben definida: crea un objecte al qual se li passa una adreça per a ubicar l'arxiu resultant de manera que se li envia un arxiu i retorna el mateix arxiu empaquetat, deixant a elecció del usuari l'opció de deixar-lo comprimit.
Si el que es vol és descomprimir, se li ha de passar l'arxiu compatible zip i retorna el contingut al directori especificat al constructor.
Per a treballar amb carpetes, ha estat necessari dissenyar una sèrie de mètodes recursius per a portar-ho a terme d'una manera eficient. La classe també disposa d'una finestra per a visualitzar el procés d'extracció dels arxius.
Aquesta classe s'utilitza per a fer diverses tasques totalment diferenciades, la naturalesa de les quals obliga a què estigui a dos llocs físics de l'aplicació: Una dins el paquet *Appactions* per a què ho utilitzi les interfícies del programa i una altra a la instal·lació de l'aplicació atès que la classe *Instalador* la utilitza per a descomprimir els arxius que van al disc dur.
Com a utilitat que queda fora de les característiques que necessita l'aplicació, aquesta classe disposa d'un mètode *main* que es pot utilitzar per a comprimir i descomprimir fitxers sense utilitzar l'aplicació principal.

#### 2.1.5.3 Classes importades

Les classes noves de la llibreria de java utilitzades aquí són les següents:

***java.util.Enumeration***

Aquesta classe s'utilitza per a representar un vector d'elements per a ser tractats.

***java.util.zip.ZipEntry***

Aquesta classe representa l'entrada d'un arxiu en format ZIP.

***java.util.zip.ZipFile***

Aquesta classe s'utilitza per a llegir les entrades d'un arxiu en format ZIP.

***java.util.zip.ZipOutputStream***

Aquesta classe proporciona un filtre de sortida per escriure arxius amb el format ZIP.

1:http://bugs.sun.com/bugdatabase/view_bug.do;jsessionid=8aed17a170173d4111e8bbb402f6:YfiG?bug_id=4244499

## 2.1.6 La classe RetornaParametres

### 2.1.6.1 Descripció

Aquesta classe té la funció d'extreure els paràmetres necessaris de l'arxiu xifrat per al posterior desxifratge, retornant el contingut net de l'arxiu xifrat sense els paràmetres de configuració i recuperant l'algorisme i el proveïdor amb què s'ha xifrat.

### 2.1.6.2 Disseny

El disseny és molt senzill: al moment de cridar a la classe, es dona com a paràmetre la ubicació i nom de l'arxiu que crea un filtre de dades per a recórrer l'arxiu i extreure la configuració. Quan s'arriba a la part on està el contingut xifrat, es retorna l'arxiu per a què sigui processat pel xifrador. A banda, la classe conté uns mètodes per a poder agafar les variables de configuració.

### 2.1.6.3 Classes importades

Les classes noves de la llibreria de java utilitzades són les següents:

***java.util.StringTokenizer***

Aquesta classe proporciona utilitats per a trencar una variable de text en trossos a partir d'un caràcter clau.

***java.io.DataInputStream***

Aquesta classe proporciona un filtre per a poder llegir dades d'un arxiu i poder-les passar després a un altre filtre per a processar-les.

## 2.1.7 La classe MonitorExtract

### 2.1.7.1 Descripció

Aquesta classe proporciona una finestra per a visualitzar el progrés que es segueix quan es descomprimeix un arxiu. Només apareix a la instal·lació del programa al disc dur.

### 2.1.7.2 Disseny

La classe és una extensió de *JPanel* que crea una finestra amb una caixa de text per a poder visualitzar dades i un botó per a poder tancar-la.

#### 2.1.7.3 Classes importades

Aquesta classe utilitza els paquets *java.awt* i *javax.swing* la seva funció ja s'ha explicat a altres classes.

## 2.1.8 La classe Manual

#### 2.1.8.1 Descripció

Aquesta classe proporciona una finestra de *java* adient per a ubicar un arxiu format *html* que mostra el manual d'usuari en línia. La idea és que l'usuari pugui disposar d'un manual mentre es té l'aplicació engegada per si hi ha algun dubte en la funcionalitat d'algun botó o opció del programa.

#### 2.1.8.2 Disseny

La classe manual mostra una finestra que s'ha de cridar a partir d'un marc pare que té un pannell on es visualitza el contingut d'una pàgina web.

Per a obtenir la màxima funcionalitat que ofereixen aquest tipus d'arxius, és necessari gestionar els hipervincles de què consta l'arxiu. *Java* proporciona una sèrie de classes per a fer-ho que es troben dins el paquet *javax.swing.text.html.*

El disseny de la finestra és equivalent al de la classe *MonitorExtract* amb la particularitat que aquí s'han de capturar els fets interns per a gestionar-los. Així s'aconsegueix una navegació fluïda per dins l'arxiu.

#### 2.1.8.3 Classes importades

Dins aquesta classe tenim les següents classes importades d'especial menció:

***java.net.URL***

S'utilitza per a convertir un camp de text que indica el nom d'un arxiu en un format adient per a que java ho reconegui com a pàgina web.

***javax.swing.event.HyperlinkEvent***

***javax.swing.event.HyperlinkListener***

***javax.swing.text.html.HTMLDocument***

***javax.swing.text.html.HTMLFrameHyperlinkEvent***

Aquestes quatre classes s'utilitzen per a mostrar i gestionar els possibles hipervincles que contingui l'arxiu en format web.

## 2.1.9 La classe IdiomaManager

### 2.1.9.1 Descripció

Aquesta classe s'encarrega de la gestió de l'aplicació i de tots els missatges de text per a què apareguin en el idioma seleccionat per l'usuari. Per a portar-ho a terme, és imprescindible que els arxius, ubicats a la carpeta *idioma* del directori de l'aplicació, existeixin i no siguin corruptes, és a dir, que no s'hagin manipulat.

### 2.1.9.2 Disseny

El disseny d'aquesta classe respon a la mateixa idea que la classe *ConfiguracioAppManager*. S'utilitza la utilitat *Properties* per a gestionar un arxiu de configuració.

### 2.1.9.3 Classes importades

No hi ha classes importades noves.

## 2.1.10 La classe EsborraCarpetes

### 2.1.10.1 Descripció

La funció d'aquesta classe és la de permetre i gestionar que s'esborri una carpeta seleccionada i els arxius que contingui. Atès que una carpeta ha d'estar buida per a esborrar-la, no es pot fer d'una manera trivial sinó que primer s'ha d'accedir als fitxers més interns i anar esborrant cap enrere.

### 2.1.10.2 Disseny

Per a efectuar aquesta cerca de fitxers interns dins una carpeta es segueix el mateix disseny utilitzat en la classe *ZipManager* per a la compressió de carpetes: mitjançant mètodes recursius, s'accedeix al fitxer més intern i s'esborra i es va cap enrere esborrant totes les carpetes quan no tinguin més arxius.

#### 2.1.10.3 Classes importades

No hi ha classes importades noves .

## 2.1.11 La classe ArxiuConfiguracio

#### 2.1.11.1 Descripció

Aquesta classe té la funció d'inserir els paràmetres necessaris com són el tipus d'algorisme i proveïdor utilitzats a l'arxiu xifrat per al posterior desxifratge. D'aquesta manera en un sol arxiu tenim el xifratge i les dades necessàries.

En una primera versió de treball l'aplicació generava dos arxius per separat, però el fet de mantenir tots dos dins un de sol és millor atès que es redueix el perill que es perdi un d'ells i sigui impossible recuperar-lo.

#### 2.1.11.2 Disseny

En aquesta classe també s'utilitza la utilitat *Properties* que ofereix *java* per a inserir les dades de configuració. Es crea l'arxiu i s'insereix els paràmetres adients, deixant una marca per a saber distingir la zona de configuració de la de xifratge. La marca que deixa és: `"<<<FI_CONFIGURACIO>>>".` D'aquesta manera es pot ubicar exactament la posició on es troben les dades xifrades.

Una vegada les dades introduïdes, es procedeix a inserir el contingut de l'arxiu xifrat i desar-lo la carpeta que li correspon.

#### 2.1.11.3 Classes importades

No hi ha classes importades noves.

## 2.1.12 La classe Instalador

#### 2.1.12.1 Descripció

Aquesta classe és responsable d'instal·lar l'aplicació al disc dur. Atès que no s'ha trobat la manera de manipular amb codi java un accés directe de *Windows*, s'ha optat per instal·lar-la

al disc C: sense possibilitat de canviar-se. En el cas que l'usuari ho vulgui canviar manualment, també s'ha de manipular el direccionament dels accessos directes del programa.

#### 2.1.12.2 Disseny

El disseny d'aquesta classe respon a un seguit de finestres d'informació i d'opció amb la finalitat de monitoritzar la instal·lació de l'aplicació al disc dur i la possibilitat de posar-hi un accés directe a l'escriptori. Per fer-ho, prèviament ha d'haver un arxiu *Data1* al directori on es troba la classe on agafa els arxius que necessita. Aquest arxiu és simplement un arxiu zip sense extensió on s'ha compactat tot el sistema d'arxius del programa i, mitjançant una instància de la classe ZipManager, es torna a descomprimir al disc C.

#### 2.1.12.3 Classes importades

No hi ha classes importades noves.

## 2.2 ASPECTES CONCRETS DE LA IMPLEMENTACIÓ

### 2.2.1 Creació dels accessos directes

Els accessos directes s'han creat sense implementar codi java, és a dir, són simplement fitxers .lnk que Windows crea quan es demana un accés directe a un arxiu. L'avantatge és que, si treballes amb l'accés del menú contextual, funciona correctament a diferència de si fos l'arxiu bat.

### 2.2.2 Inserir un accés directe al menú contextual

Per a crear un accés directe al menú contextual s'ha de saber quin és el directori de treball de l'usuari. Una vegada obtingut, s' insereix dins la carpeta *SendTo* .

Una manera fàcil i eficient d'obtenir aquest directori la proposa java amb la classe *System.* Aquesta classe, que es troba al paquet *java.lang*, ofereix un conjunt de mètodes interessants. Un d'ells, el mètode *getProperty(String key),* dona l'opció de saber des del directori on està instal.lat java fins el nom de l'usuari i el seu directori de treball.

### 2.2.3 Canvi de les polítiques de seguretat

Els fitxers de polítiques de seguretat de java és troben a *%java.home%\lib\security\ i són: local_policy.jar* i *US_export_policy.jar*. Atès que a molts països hi ha lleis molt restrictives a l'hora d'utilitzar algorismes de xifratges robusts, aquests fitxers no deixen utilitzar el 3DES perquè té una clau de 128 bits. Per solucionar-ho, s'han de canviar aquests dos arxius per uns altres que no limitin la longitud de clau. En versions anteriors del programa, s'havien de canviar a mà. S'ha optat per fer-ho d'una manera transparent a l'usuari i quan l'aplicació s'engega, automàticament, els canvia deixant-ho tot tal i com estava quan es tanca.

### 2.2.4 Inserir el proveïdor IAIK a l'arxiu de seguretat de java

L'aplicació pot treballar amb dos proveïdors de seguretat diferents: SUN JCE i IAIK. El primer ja es troba dins l'arxiu de seguretat. IAIK, però, per a què java el reconegui com a tal, s'ha d'inserir. Aquest fitxer està ubicat a *%java.home%\lib\security* i es diu *java.security.*
També s'ha optat per fer la inserció d'una manera transparent a l'usuari i l'aplicació només l'instal·la quan es configura un algorisme que el necessiti.

## 2.3 REQUERIMENTS DE L'APLICACIÓ

L'aplicació és utilitzable en tots aquells ordinadors que disposin de Windows 2000 o Windows XP, en altres sistemes operatius, no és pot assegurar que funcioni correctament. Les proves d'execució del programa han estat portades a terme en ordinadors que disposaven d'un d'aquests dos sistemes operatius, no és pot garantir, doncs, la utilització d'aquest programa en aquells ordinadors que disposin d'una versió anterior o posterior a aquests sistemes.

A més a més dels requeriments de sistema operatiu, també és necessari que l'ordinador disposi d'una maquina virtual de JAVA. Aquesta aplicació ha estat dissenyada amb la versió 1.4.2 de JAVA. No es garanteix el funcionament correcte d'aquest programa en aquells ordinadors que, malgrat disposar del sistema operatiu adient, tinguin instal·lada una versió anterior de màquina virtual.

Per tal d'instal·lar l'aplicació és indispensable almenys d'1,5 MB. d'espai al disc i 32 MB de memòria RAM. El programa en sí utilitza uns 18 MB de memòria de treball en la seva execució i no fa cap despesa quan l'aplicació està resident.

## 2.4 MANUAL D'INSTAL.LACIÓ

L'arxiu *Instalar.bat* ens permet la instal·lació del programa. Si el seleccionem i hi entrem, apareix una finestra, sempre en català, que ens demana que confirmem si volem instal·lar *cryptoStar.*


*Figura 3: Finestra d'instal·lació. FONT: CriptoStar v1.0*

En cas de confirmar-ho i després de seleccionar SI, apareix una altra finestra per tal de determinar la ubicació física de l'aplicació dintre de l'ordinador.
Per defecte i sense possibilitat de canviar-ho, la ubicació del programa serà a C:\cryptoStar\.

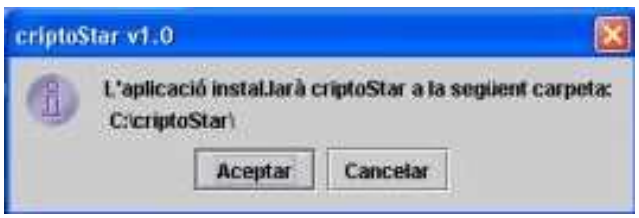



*Figura 4: Finestra d'instal·lació 2 . FONT: CriptoStar v1.0.*

Si premem el botó *Aceptar*, s'obrirà una altra finestra que mostrarà el procés d'instal·lació. En el cas que al disc C: ja hi hagi una carpeta amb el mateix nom, és a dir, que C:\cryptoStar ja existeixi, el procés d'instal·lació demanarà confirmació abans d'instal·lar el programa.

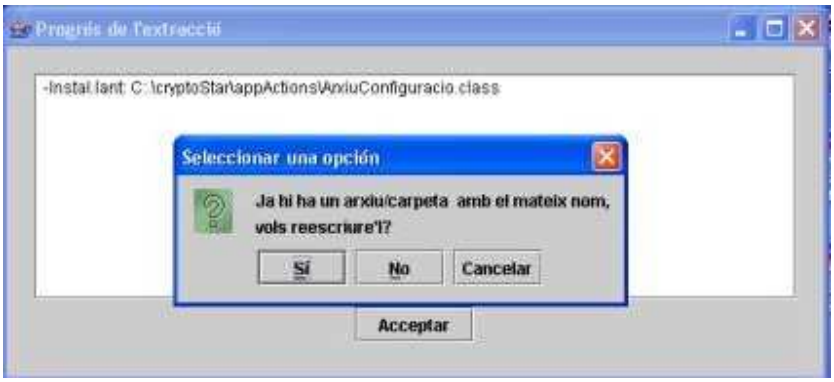



*Figura 5: Finestra d'instal·lació 3. FONT: CriptoStar v1.0*

Una vegada confirmat que es vol continuar el procés, l'aplicació copia els arxius necessaris al disc dur. Quan ha acabat de copiar-los-hi, demana si volem instal·lar un accés directe a l'aplicació a l'escriptori.

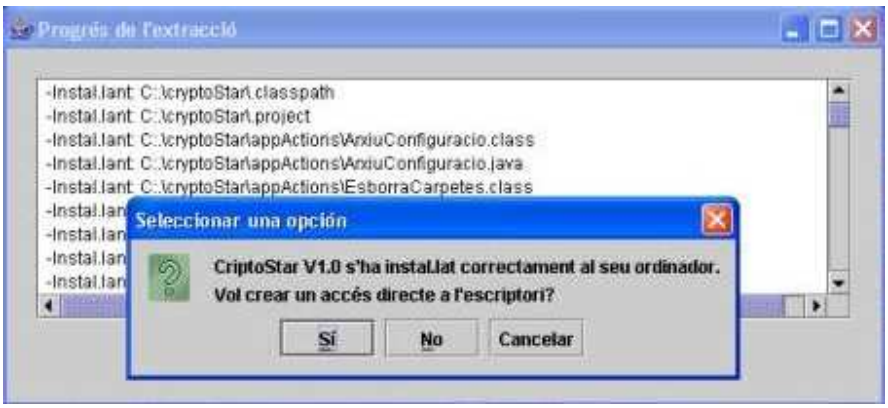



*Figura6 : Finestra d'instal·lació 4. FONT: CriptoStar v1.0*

Una vegada acceptada o denegada l'opció de l'accés directe, el procés ha estat portat a terme.

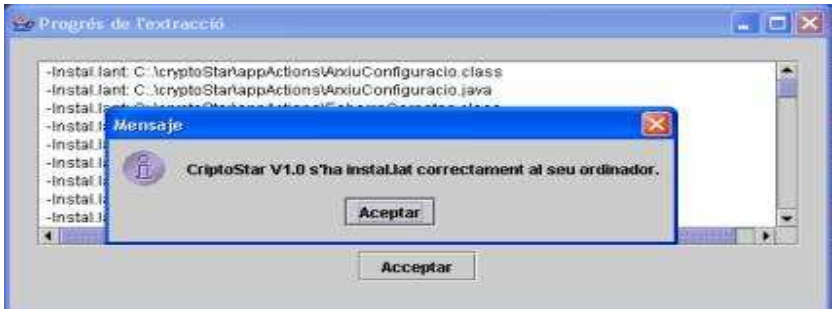



*Figura 7: Finestra d'instal·lació 5. FONT: CriptoStar v1.0*

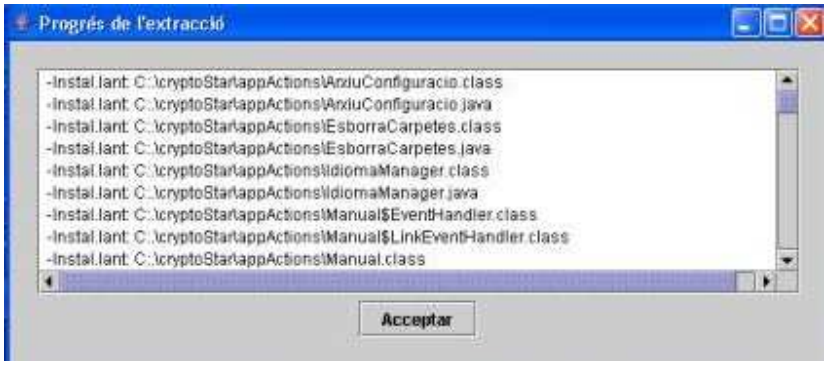



*Figura 8: Finestra d'instal·lació 6. FONT: CriptoStar v1.0*

Una vegada està l'aplicació al disc dur, per engegar-la, simplement s'ha de pitjar sobre l'accés directe de l'escriptori anomenat criptoStar, si s'ha volgut instal·lar, o bé sobre el que es troba al directori C:\criptoStar\.

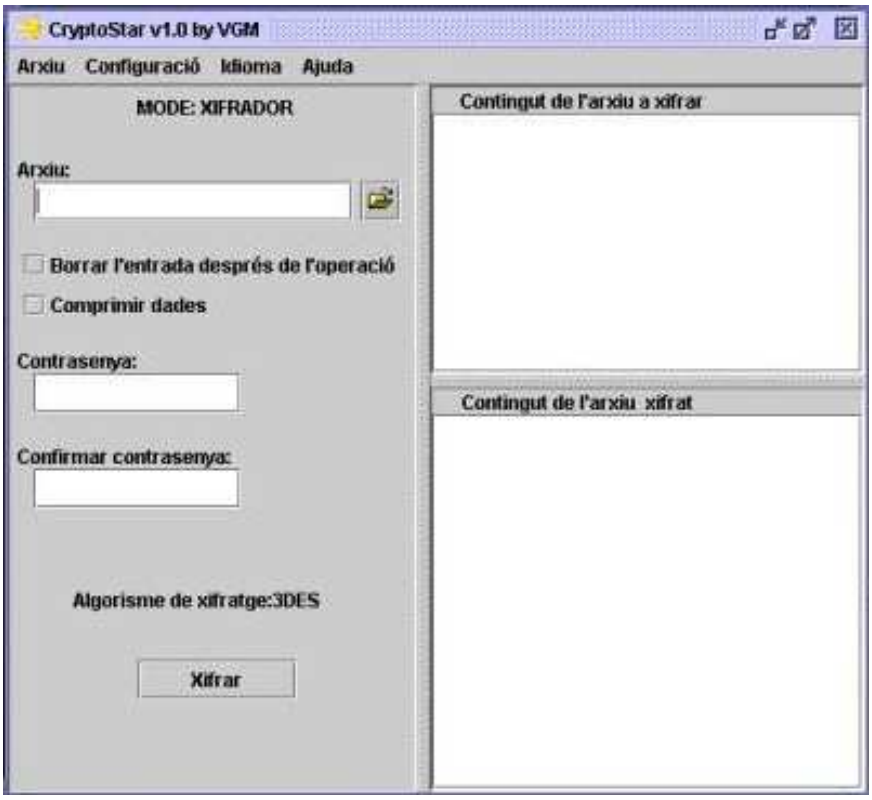
## 2.5 MANUAL D'USUARI

### 2.5.1 Introducció

*CriptoStar* és un programa que permet xifrar fitxers individuals o agrupats en carpetes mitjançant el xifratge basat en contrasenya. Per tant, per a poder xifrar, és necessari introduir una contrasenya. Una vegada xifrat l'arxiu o carpeta, el seu contingut és il·legible per qualsevol usuari dotant al xifratge d'una robustesa variable en funció de la llargària de la contrasenya i de l'algorisme de xifratge emprat. El mateix programa pot desxifrar el contingut de l'arxiu sempre i quan es posi la contrasenya correcta, en cas contrari, no es podrà restaurar el contingut original. Per tant, només els coneixedors de la contrasenya podran accedir al contingut real dels fitxers xifrats.

### 2.5.2 La interfície principal

La interfície principal està dividida en dues àrees i un menú situat a la part superior.
L'àrea ubicada a l'esquerra demana els paràmetres necessaris per al funcionament del programa mentre que l'àrea situada a la part dreta està pensada per a visualitzar el contingut del fitxer abans i després del seu xifratge.
L'espai de la pantalla utilitzat per cada una de les dues àrees és adaptable a les necessitats de l'usuari de manera que, una vegada entrats els paràmetres, es pot ampliar l'àrea de visualització tan com sigui necessari.
Per tal de modificar l'abast de cada zona només s'ha d'arrossegar la barra que separa les dues àrees mantenint el botó esquerre del ratolí pitjat.




ÀREA DE PARÀMETRES ÀREA DE VISUALITZACIÓ

*Figura 9: Interfície del menú principal. Mode xifrador. FONT: CriptoStar v1.0*

#### 2.5.2.1 Menú: Descripció i funcions

El Menú està situat a la part superior de la interfície principal. Aquest menú contempla quatre opcions diferents: Arxiu, configuració, idioma i ajuda.

##### 2.5.2.1.1 Arxiu

L'opció *Arxiu* contempla quatre possibilitats inherents a l'aplicació i al seu funcionament: Xifrar, desxifrar, deixar-lo resident i sortir.
Les opcions *xifrar* i *desxifrar* indiquen el mode de treball del programa; l'opció *deixar-lo resident* permet sortir de la interfície principal mantenint un accés directe al menú contextual dels arxius per tal d'utilitzar-lo més tard i l'opció *sortir* permet sortir de l'aplicació.

##### 2.5.2.1.2 Configuració

L'opció *configuració* permet que l'usuari seleccioni el tipus d'algorisme i el proveïdor de seguretat que s'ha d'aplicar al xifratge. Aquesta opció només està habilitada per al mode de

treball de xifratge atès que el mode de desxifratge ja inclou dins el mateix arxiu l'algorisme i el proveïdor. Aquesta opció, doncs, resta inhabilitada en el mode desxifratge.

L'opció *configuració* inclou les cinc opcions següents: *DES SUN*, *DES IAIK*, 3DES SUN*, 3DES IAIK* i *RC2 IAIK*.

L'opció *DES SUN* xifrarà l'arxiu o carpeta amb l'algorisme de xifratge *DES* del proveïdor de seguretat *SUN JCE.*

L'opció *DES IAIK* xifrarà l'arxiu o carpeta amb l'algorisme de xifratge *DES* del proveïdor de seguretat *IAIK.*

L'opció 3*DES SUN* xifrarà l'arxiu o carpeta amb l'algorisme de xifratge triple*DES* del proveïdor de seguretat *SUN JCE.*

L'opció *3DES IAIK* xifrarà l'arxiu o carpeta amb l'algorisme de xifratge triple*DES* del proveïdor de seguretat *IAIK.*

L'opció *RC2 IAIK* xifrarà l'arxiu o carpeta amb l'algorisme de xifratge *RC2* del proveïdor de seguretat *IAIK.*

L'algorisme que permet un xifratge més robust, és a dir, més difícil de desxifrar, és el 3DES i el que permet un xifratge més feble és el RC2.

#### 2.5.2.1.3 Idioma

L'opció Idioma permet canviar l'idioma en què apareixen els missatges de l'aplicació. Tots els missatges del programa poden ser llegits en català ( opció *Català*), en castellà ( opció *Español*), en anglès ( opció English) i en francès ( opció *Français*).

#### 2.5.2.1.4 Ajuda

L'opció *Ajuda* inclou dues opcions: l'opció *Manual d'usuari* i l'opció *Sobre cryptoStar.*

L'opció *Manual d'usuari* permet l'accés a la descripció i les funcions del programa mentre que l'opció *Sobre cryptoStar* permet l'accés a les dades pròpies de l'aplicació ( dades de l'autor, de l'any de realització etc.).

### 2.5.2.2 Àrea de paràmetres

L'*àrea de paràmetres* demana els paràmetres necessaris per tal d'executar el xifratge o el desxifratge dels arxius o les carpetes.

Per al xifratge d'arxius o de carpetes, la zona de paràmetres presenta set opcions mentre que per al desxifratge en presenta només sis. A més a més, en el mode desxifratge hi ha una opció que apareix per defecte mentre que en el mode de xifratge la mateixa opció és seleccionada per l'usuari.
A continuació s'inclou una breu descripció del funcionament dels paràmetres en cada un dels dos modes de treball.

#### 2.5.2.2.1 Mode xifratge

En el mode de xifratge la primera de les opcions, *Mode*, apareix com a *XIFRADOR*.
La segona opció *Arxiu* permet seleccionar l'arxiu o la carpeta a xifrar que pot estar ubicada en qualsevol unitat de l'ordinador. Per tal de seleccionar-la, hi ha dues possibilitats:

1. Escriure directament el nom de l'arxiu o la carpeta a xifrar.
2. Prémer la icona carpeta ( situada al costat de la casella en blanc) que actua com a un cercador a partir del qual seleccionarem l'arxiu o la carpeta de qualsevol unitat.

La tercera opció és *Borrar l'entrada després de l'operació.* Si es selecciona la casella al costat d'aquesta opció, l'aplicació borra l'arxiu original seleccionat per al xifratge.
La següent opció és *Comprimir dades.* Aquesta opció compacta o comprimeix l'arxiu xifrat. Si marquem la casella al costat d'aquesta opció, l'arxiu final queda, a més a més de xifrat, comprimit.
Les dues opcions següents *Contrasenya* i *Confirmar contrasenya* són les responsables de mantenir la seguretat del missatge xifrat on només tindrà accés l'usuari coneixedor de la contrasenya.
La següent opció *Algorisme de xifratge* només és modificable des de l'opció de menú *Configuració* i per a l'opció *xifrador*.
Una vegada determinats els paràmetres de totes les opcions, s'ha de prémer la casella *Xifrar* per tal de començar el procés.

#### 2.5.2.2.2 Mode desxifratge

En el mode de desxifratge la primera de les opcions, *Mode*, apareix com a *DESXIFRADOR*.
La segona opció *Arxiu* permet seleccionar l'arxiu o la carpeta a desxifrar que pot estar ubicada en qualsevol unitat de l'ordinador. Per tal de seleccionar-la, hi ha dues possibilitats:

1. Escriure directament el nom de l'arxiu o la carpeta a desxifrar.

2. Prémer la icona carpeta ( situada al costat de la casella en blanc) que actua com a un cercador a partir del qual seleccionarem l'arxiu o la carpeta de qualsevol unitat.

La tercera opció és *Borrar l'entrada després de l'operació.* Si es selecciona la casella al costat d'aquesta opció, l'aplicació borra l'arxiu original seleccionat per al desxifratge.
La següent opció és *Comprimir dades* no està habilitada en el mode desxifrador.
Les dues opcions següents *Contrasenya* i *Confirmar contrasenya* són les responsables de mantenir la seguretat del missatge a on només tindrà accés per tal de desxifrar-lo l'usuari coneixedor de la contrasenya.
La següent opció *Algorisme de xifratge* ve determinada pel mode de xifratge de l'arxiu a desxifrar. Només s'habilita una vegada acceptat l'arxiu que és procedirà a desxifrar.
Una vegada determinats els paràmetres de totes les opcions, s'ha de prémer la casella *Desxifrar* per tal de començar el procés.

#### 2.5.2.3 Àrea de visualització

L' *Àrea de visualització* està dividida en dos camps principals:

1. *Contingut de l'arxiu a xifrar.*
2. *Contingut de l'arxiu xifrat.*

Cada camp ubica l'arxiu xifrat i l'arxiu sense xifrar independentment del mode de treball.

### 2.5.3 La interfície del menú contextual

Aquesta opció es per tal de treballar d'una manera més ràpida i eficient. En lloc d'entrar a l'aplicació i cercar l'arxiu des de l'aplicació, en aquest cas es treballa des de l'arxiu i l'opció de què disposa l'arxiu al seu menú contextual dins l'opció *Enviar a*.
Segons l'extensió de l'arxiu, aquesta interfície es posa automàticament en mode *xifrador* o *desxifrador*. Si l'extensió de l'arxiu és *crs*, es posa en opció *desxifrar.* En qualsevol altra extensió, es posa en mode *xifrar.*

#### 2.5.3.1 Mode xifrar

En el mode *xifrar*, el *proveïdor* i *l'algorisme* es selecciona de les opcions escollides a la interfície principal.

El *fitxer/carpeta* és el fitxer des d'on s'ha cercat l'opció al menú contextual.

Per tant, les opcions a omplir són les dues referents a la contrasenya ( *Contrasenya* i *verificació*) i les opcions a seleccionar són *borrar l'entrada després de l'operació* i *comprimir dades.*

Si es selecciona la casella al costat de *borrar l'entrada després de l'operació*, l'aplicació borra l'arxiu original seleccionat per al xifratge.

L'opció *comprimir dades* compacta o comprimeix l'arxiu xifrat. Si marquem la casella al costat d'aquesta opció, l'arxiu final queda, a més a més de xifrat, comprimit.

La part inferior de la casella inclou quatre opcions: *Torna a l'aplicació, acceptar, cancel·la* i *tancar l'aplicació.*

L'opció *Torna a l'aplicació* obre la pantalla principal del programa. Això ens permet canviar algun dels paràmetres no modificables des de la interfície del menú contextual.

L'opció *acceptar* permet que es posi en marxa el procés de xifratge amb els paràmetres seleccionats si la contrasenya és verificada mentre que l'opció *cancel·la* l'atura.

L'opció *tancar l'aplicació* tanca l'aplicació.

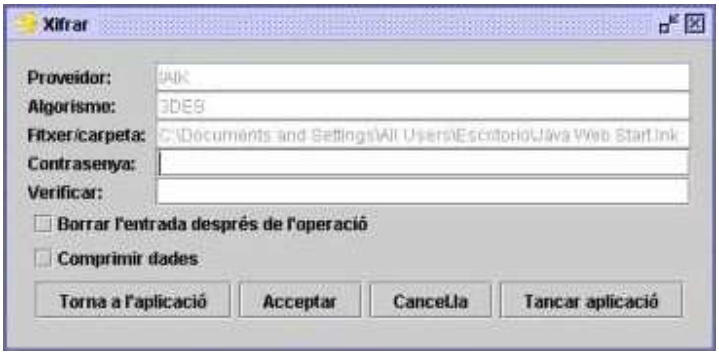



*Figura10: Interfície del menú contextual. Mode xifrar. FONT: CriptoStar v1.0*

### 2.5.3.2 Mode desxifrar

En el mode *desxifrar*, el *proveïdor* i *l'algorisme* es selecciona de les opcions que porta l'arxiu xifrat.

El *fitxer/carpeta* és el fitxer des d'on s'ha cercat l'opció al menú contextual.

Per tant, les opcions a omplir són les dues referents a la contrasenya ( *Contrasenya* i *verificació*) i l'opció a seleccionar és *borrar l'entrada després de l'operació.*

Si es selecciona la casella al costat de *borrar l'entrada després de l'operació*, l'aplicació borra l'arxiu original seleccionat per al xifratge.

L'opció *comprimir dades* no està habilitada.

La part inferior de la casella inclou quatre opcions: *Torna a l'aplicació, acceptar, cancel·la* i *tancar l'aplicació.*

L'opció *Torna a l'aplicació* obre la pantalla principal del programa. Això ens permet canviar algun dels paràmetres no modificables des de la interfície del menú contextual.

L'opció *acceptar* permet que es posi en marxa el procés de desxifratge amb els paràmetres seleccionats si la contrasenya és verificada mentre que l'opció *cancel·la* l'atura.

L'opció *tancar l'aplicació* tanca l'aplicació.

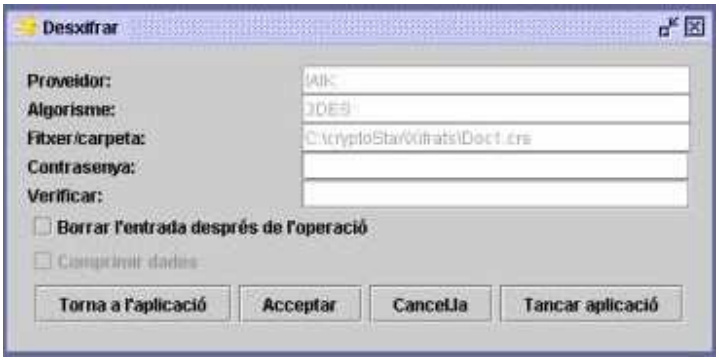



*Figura 11: Interfície del menú contextual. Mode desxifrar FONT: CriptoStar v1.0*

## 2.6 DESCRIPCIÓ DE LES PROVES DE FUNCIONALITAT

Una vegada instal·lada l'aplicació al disc dur, s'ha d'obrir a partir de l'accés directe creat a l'escriptori. La interfície que es pot veure és:

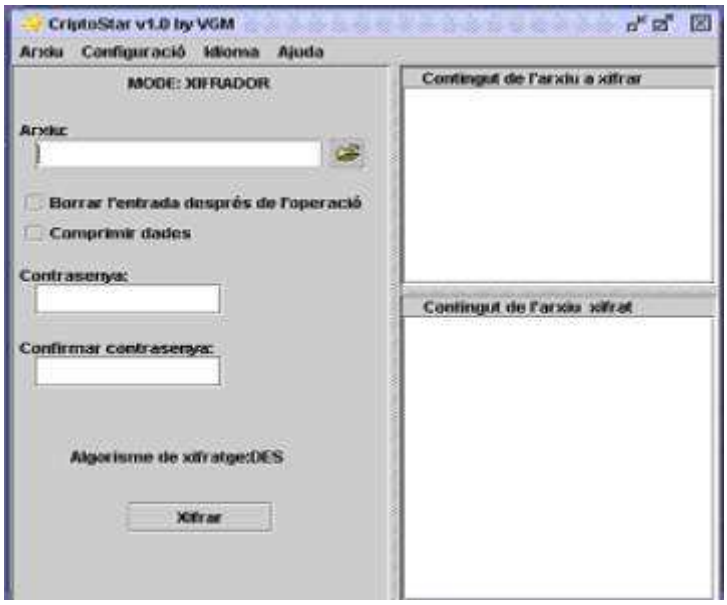



*Figura 12: Interfície principal. FONT: CriptoStar v1.0*

L'aplicació per defecte es troba en mode xifratge, es pot canviar l'algorisme de xifratge per un de més robust, per exemple 3DES IAIK:

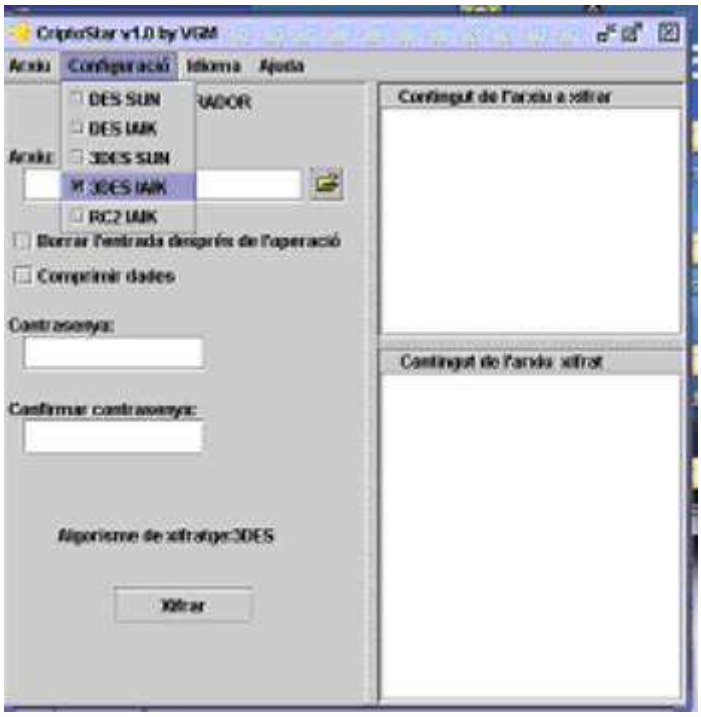



*Figura 12: Interfície del menú contextual. Configuració FONT: CriptoStar v1.0*

Una vegada seleccionat l'algorisme podem xifrar un arxiu, per fer-ho podem escriure-ho directament a la casella habilitada o bé es pot utilitzar el cercador. Si s'utilitza el cercador:

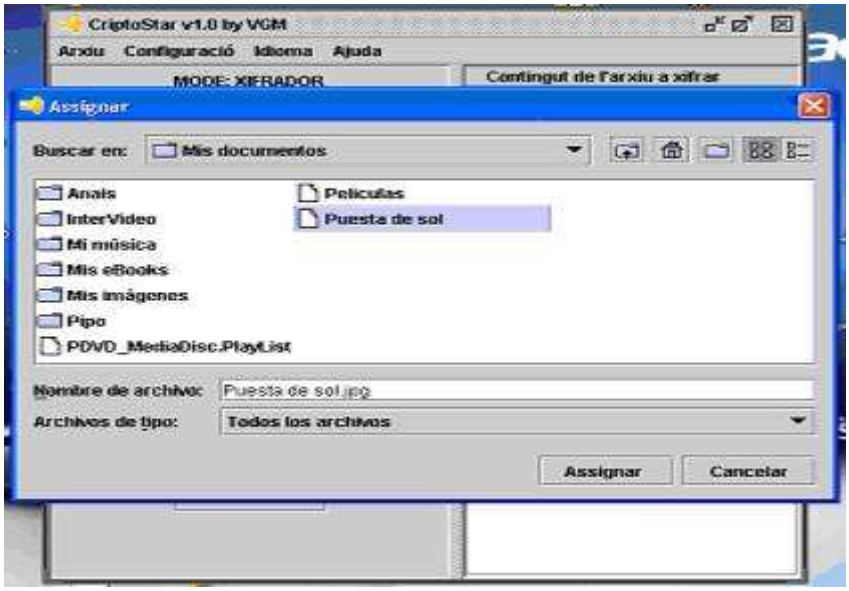



*Figura 13: Cerca. FONT: CriptoStar v1.0*

Es selecciona l'arxiu *Puesta de sol.jpg* per a xifrar. El seu contingut se pot visualitzar a la part dreta del programa a la zona habilitada per a això. Es selecciona les opcions d'esborrar l'entrada i comprimir les dades les quals avisen cadascuna amb missatges d'avís de les opcions escollides:

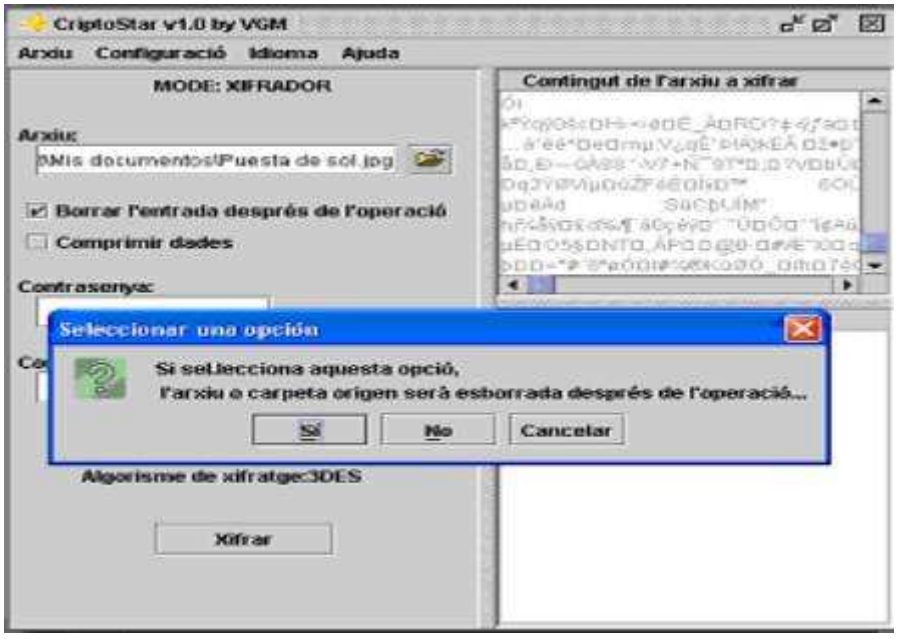



*Figura 14: elecció d'opció FONT: CriptoStar v1.0*

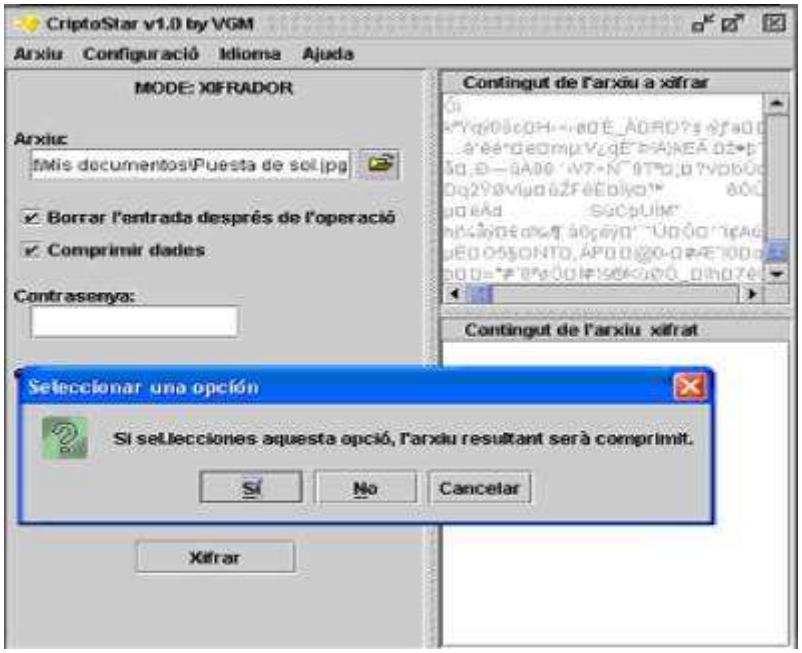



*Figura 15: elecció d'opció 2 FONT: CriptoStar v1.0*

Una vegada seleccionades aquestes opcions que no són obligades per al funcionament correcte de l'aplicació, s'introdueix la contrasenya. Es posa la confirmació. Com a exemple d'una possible errada que és capaç de controlar, es posen dues contrasenyes diferents a cada camp per a que ens mostri un missatge d'error quan es prem el botó acceptar:

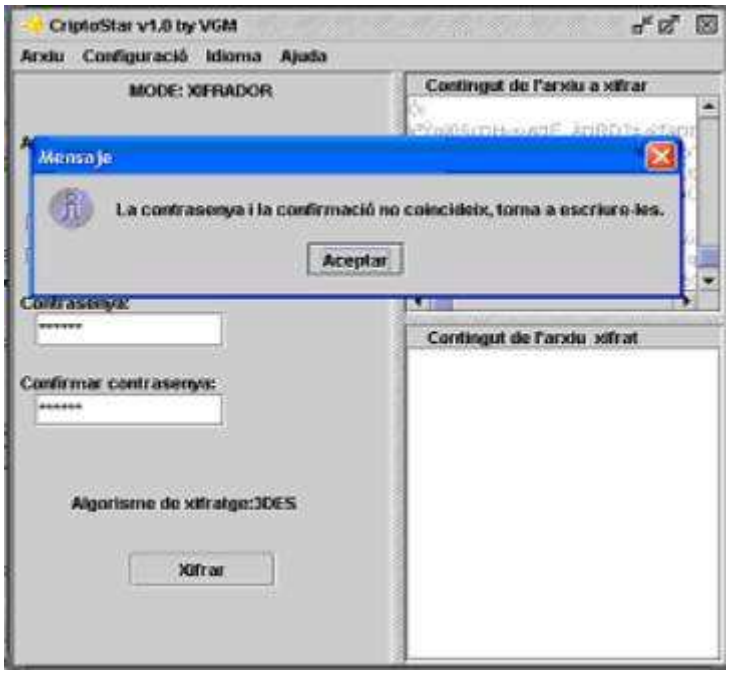



*Figura 16: No coincidència de contrasenya i confirmació. FONT: CriptoStar v1.0*

Si es posen les contrasenyes bé i es pitja acceptar al final del xifratge surt un missatge corroborant-ho:

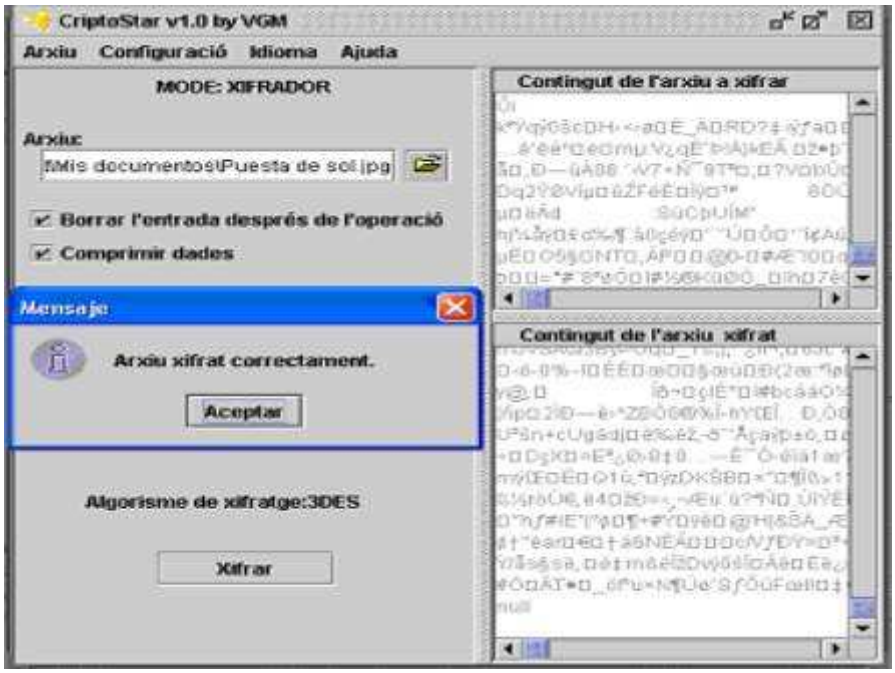



*Figura 17: Coincidència de contrasenya i confirmació. FONT: CriptoStar v1.0*

És pot observar que si el xifratge ha tingut èxit, el contingut de l'arxiu xifrat es visualitza a la caixa de text inferior. A la carpeta on estava l'arxiu, pitjant una altra vegada el botó de cerca, es pot observar que ja no hi és i a més, ara l'arxiu resultant dels xifratge ocupa menys espai que abans:

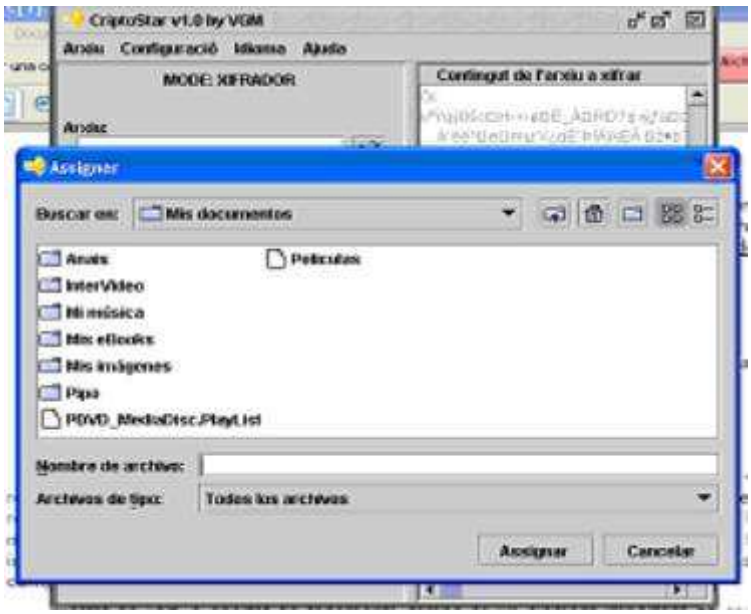



*Figura 18: Assignació. FONT: CriptoStar v1.0*

L'arxiu ja està xifrat. Per a mostrar el funcionament del programa quan està resident en memòria, es desxifra l'arxiu xifrat mitjançant l'accés directe que es crea al menú contextual. Primer es posa l'aplicació resident. Per fer-ho, premem sobre l'opció habilitada per a això:


*Figura 19: Arxiu. Resident. FONT: CriptoStar v1.0*

Una vegada ha desaparegut l'aplicació, es cerca l'arxiu a desxifrar. Com que el programa té una carpeta per a guardar els arxius xifrats, es va al directori d'instal·lació del programa i s'accedeix a la carpeta xifrats, apareix l'arxiu:

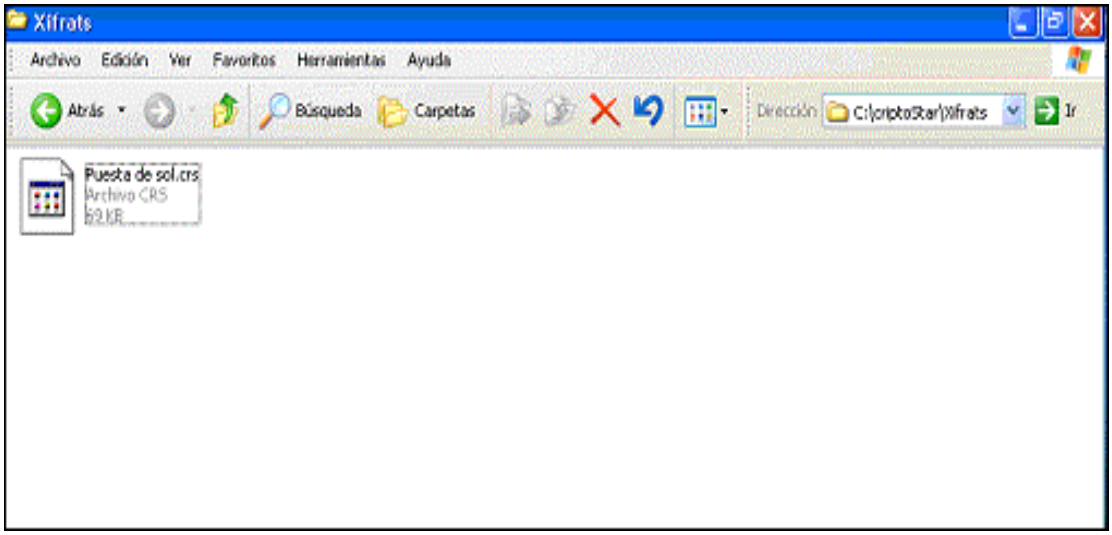



*Figura 20: Xifrats. FONT: CriptoStar v1.0*

Una vegada apareix l'arxiu cercat, amb el botó dret del ratolí es selecciona i dins el submenú *enviar a....es* troba l'accés directe de l'aplicació:

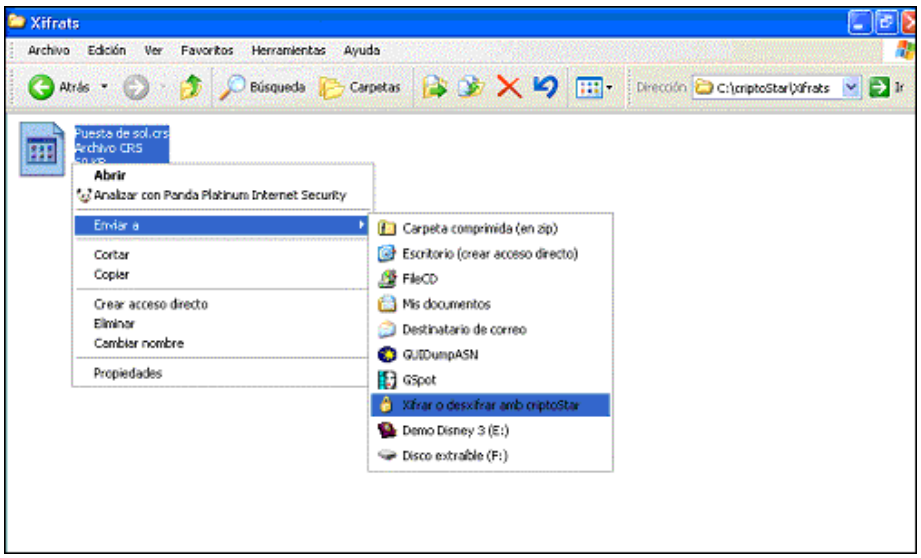



*Figura 21: Xifrar o desxifrar. FONT: CriptoStar v1.0*

Una vegada pitjat s'obre la interfície resident, es pot observar com en els tres primers camps, es visualitzen l'algorisme i proveïdor amb què s'han xifrat l'arxiu i l'arxiu que s'ha seleccionat:

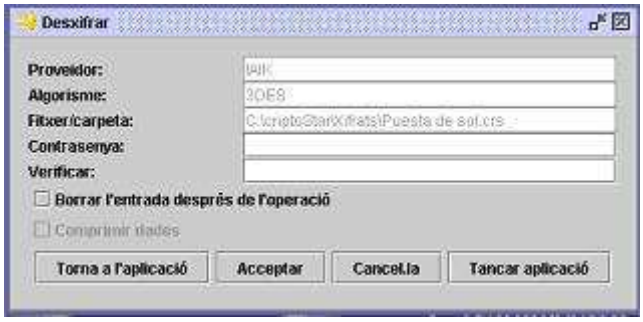



*Figura 22: Desxifrar. FONT: CriptoStar v1.0*

Si s'introdueix una contrasenya errònia un missatge avisa que no és correcta:


*Figura 23: Error. FONT: CriptoStar v1.0*

Es posa la contrasenya correcta i es selecciona l'opció d'esborrar l'arxiu .Un missatge ho indicarà:


*Figura 24: Arxiu desxifrat correctament. FONT: CriptoStar v1.0*

Ara, si es va a la carpeta desxifrats, es pot observar com l'arxiu s'ha desxifrat correctament i a banda, s'ha esborrat l'arxiu xifrat:

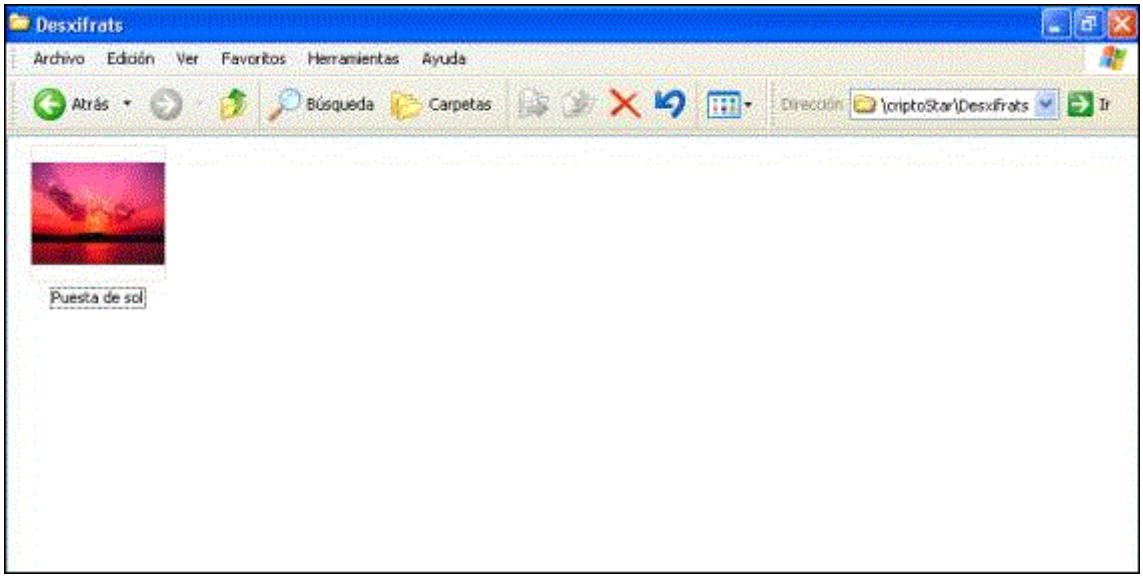



*Figura 25: Desxifrats. FONT: CriptoStar v1.0*

A la interfície principal, es canvia l'idioma de l'aplicació, per fer-ho, del menú del programa es selecciona l'opció idioma i es selecciona *english*:

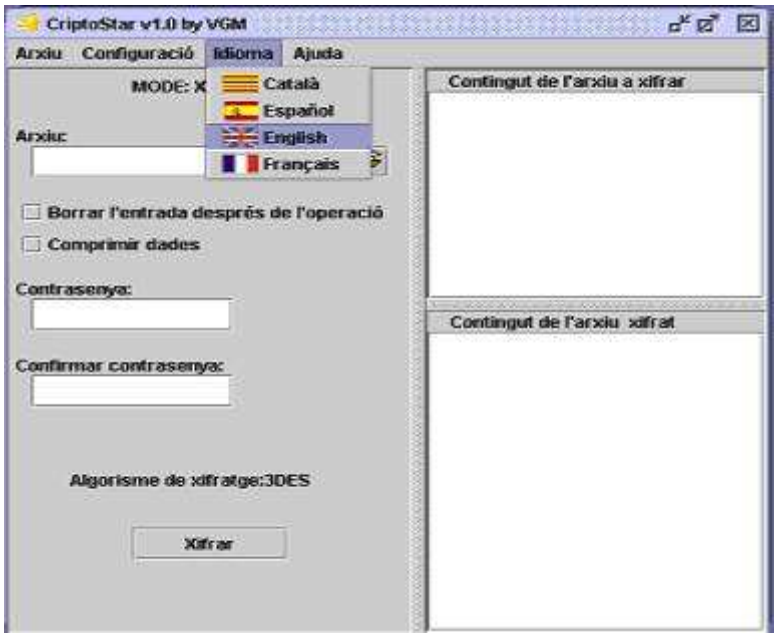



*Figura 26: Idioma. FONT: CriptoStar v1.0*

I es pot observar com l'idioma de l'aplicació ha canviat:

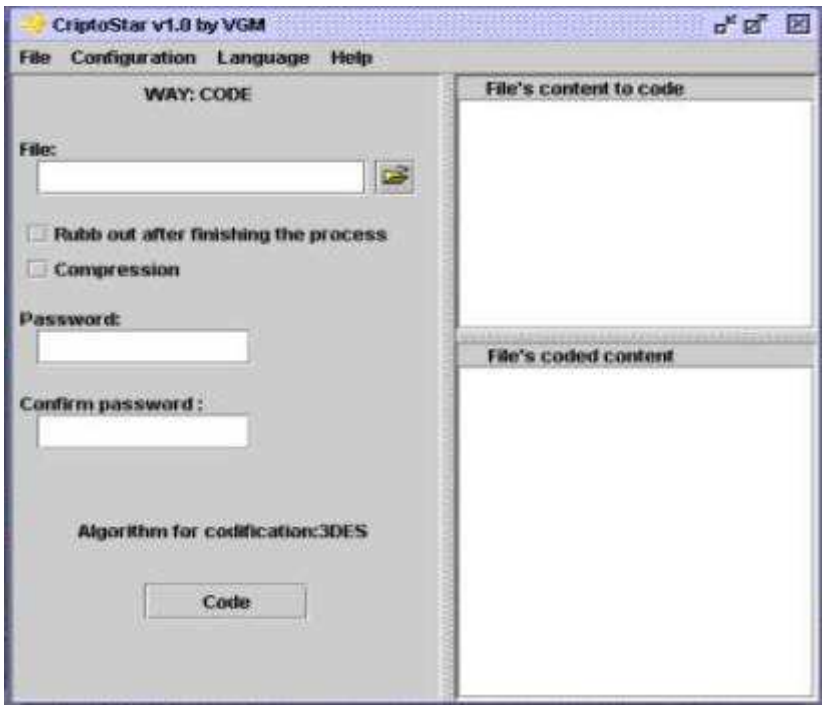



*Figura 27: Idioma2. FONT: CriptoStar v1.0*

# 3. CONCLUSIONS I LÍNIES DE TREBALL FUTURES

Aquest programa representa una primera línia de treball en un camp que es pronostica com imprescindible per a la futura eficàcia i garantia d'aquelles noves tecnologies que utilitzen internet i connexions a xarxes.
La única manera de garantir el desenvolupament de totes aquelles activitats vinculades a la xarxa és aquella que permet mantenir la privacitat i la seguretat de les dades emmagatzemades i/o transmeses.
Només amb sistemes codificadors i descodificadors que garanteixin uns mínims de seguretat, les empreses i els particulars poden permetre's aquests tipus de comunicació tan exposada a atacs i intromissions no autoritzades des de l'exterior.

Aquest programa intenta cobrir dos aspectes imprescindibles: D'una banda, la seva execució és fruit d'uns coneixements en informàtica que permeten l'aplicació de les especificitats pròpies que el fan inaccessible per a la majoria d'usuaris. D'altra banda, el disseny intenta ser el més intuïtiu i simple possible per tal de fer-lo funcional i útil per a tot tipus de públic.
Al mateix temps, per a garantir aquest públic, es presenta una aplicació accessible des de diferents llengües que són conegudes per una part important de la població susceptible d'utilitzar un tipus de programa com aquest ( És a dir, de la població amb accés a Internet). Precisament, la proposta en vàries llengües seria millorable: Es podria garantir una traducció més acurada de mà d'un especialista en llengües i es podria incloure un manual d'usuari en les diferents llengües en què es proposen els missatges. A més a més, seria modificable el seu format per tal de facilitar l'accés.

Finalment, també es podrien introduir més claus i proveïdors per a garantir una opcionalitat i un grau de robustesa més alt.

# 4. BIBLIOGRAFIA

**Ceballos, F.J.** (2000) *Java2 Curso de programación*. Madrid: RA-MA Editorial

**Cuenca, M.J.; Marco, M.J.; Nicolau, F.** (2005) *Competència Comunicativa per a professionals de la informàtica*. Barcelona: UOC

**Domingo, J.; Herrera, J. ; Rifà, H.** (2004) *Criptografia*. Barcelona: UOC

**Moldes, F.J.**(2005) *Java 2 v5.0.* Madrid: Anaya Multimedia

**Zukowski, J.** (1999) *Programación en Java2.* Madrid: Anaya Multimedia

URL's:

http://www.gnupg.org/gph/es/manual/c190.html
web per obtenir informació del xifrat simètric

http://www.uv.es/~sto/cursos/seguridad.java/html
web per obtenir informació sobre seguretat informàtica en java

http://www.eclipse.org
web del IDE eclipse.

http://java.sun.com/
web de sun.

http://www.javahispano.org
web on es pot trobar talls de codi i exemples en java.

http://www.lawebdelprogramador.com
web on es pot trobar talls de codi i exemples en java.

http://www.codigofuente.net
web on es pot trobar talls de codi i exemples en java.

http://www.solotutoriales.com
web on hi ha tutorials de java

ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.pdf
norma pkcs5 accessible des de la web.

ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12v1.pdf
norma pkcs12 accessible des de la web.