Xifratge d’arxius

TFC – Seguretat Informàtica

Autor: José Cabello Sánchez
ETIG - UOC

Consultor: Antoni Martínez Ballesté

10 de gener de 2006

TAULA DE CONTINGUTS

TAULA DE FIGURES

1 Especificació del problema

1.1 Temàtica del treball

El tema, relacionat amb la seguretat informàtica, escollit pel desenvolupament del treball final de carrera és el de **xifratge d'arxius**.

La motivació per escollir aquesta temàtica neix de la lectura del llibre *"The Code Book"* de Simon Singh durant l'estiu de l'any 2004. En aquest llibre es narra una història de la criptografia des dels seus inicis fins a l'actualitat. Simon Singh descriu diferents tècniques de codificació amb detall i les acompanya amb algunes de les famoses històries que les tenen com a protagonistes singulars, com per exemple les relacionades amb la màquina Enigma utilitzada pels nazis en les seves comunicacions segures i com l'exèrcit polonès i posteriorment l'anglès van aconseguir desvetllar el seu funcionament i per tant desxifrar les comunicacions, és a dir, conèixer el secret de l'enemic.

La lectura d'aquest llibre va ser el primer pas que em va portar a escollir i cursar l'assignatura optativa de Criptografia durant el semestre de primavera del curs 2004/2005. Vaig trobar molt interessant aquesta assignatura la qual cosa em va fer donar un pas endavant i decidir que el treball final de carrera giraria al voltant de la seguretat informàtica per tal de poder aprofundir en l'ús de les tècniques criptogràfiques per al xifratge d'arxius.

1.2 Objectiu

L'objectiu d'aquest treball és, per tant, la implementació d'una utilitat per xifrar fitxers amb claus basades en contrasenyes. En concret, l'usuari introduirà una contrasenya que li permetrà xifrar i, posteriorment desxifrar, un determinat arxiu o carpeta. La clau per xifrar i desxifrar es basa en la contrasenya introduïda de forma que el sistema sigui prou segur.

El llenguatge de programació utilitzat per desenvolupar aquesta aplicació serà Java, i les llibreries criptogràfiques que s'utilitzaran seran les proporcionades per Sun Microsystems en la versió 5 del seu software (JDK 1.5.0), de forma que es minimitzi la dificultat d'instal·lació del programari.

Un altre objectiu és utilitzar tècniques de xifratge estàndard, fent possible la modificació del número i tipus d'algorismes implementats de forma ràpida i sense haver de modificar gaires línies de codi.

2 Fonaments o estat de l'art

2.1 Seguretat informàtica

Avui dia, les aplicacions informàtiques permeten als usuaris desenvolupar tota una sèrie de feines relacionades amb el món empresarial. Aquestes aplicacions informàtiques usualment formen part d'un sistema, com per exemple una empresa. Aquests sistemes s'anomenen "sistemes segurs" si aconsegueixen fer més difícil a la gent tot allò que no estan autoritzats a fer.

Els sistemes segurs estan dissenyats pensant que el cost de trencar algun component del sistema sobrepassi els possibles beneficis. Aquests beneficis són generalment diners, control o informació que es pot vendre per diners. Normalment, la seguretat d'un sistema serà proporcional als recursos que ha de protegir.

De totes maneres, cal dir que el terme *"sistema segur"* no és gaire realista, ja que no existeix cap sistema que sigui absolutament segur. Qualsevol sistema es pot trencar, invertint més o menys temps i diners, però si es poden dissenyar sistemes que siguin prou segurs per tal de desanimar qualsevol intent de trencament.

Dissenyar i implementar una aplicació informàtica segura implica fer un balanç a tres bandes entre el cost de "perdre" les dades per un atac informàtic, el cost efectiu de desenvolupar una aplicació segura i els canvis en la facilitat d'ús de la nova aplicació. Així per exemple no té cap sentit dissenyar una aplicació amb un cost de milions d'euros si el cost d'una pèrdua d'informació es pot valorar només en milers d'euros o si el rendiment dels treballadors disminueix a causa de la dificultat de funcionament de la nova aplicació.

La ràpida implantació de la xarxa Internet ha estat eclipsada per la no menys ràpida aparició de delictes informàtics lligats a les aplicacions que n'utilitzen aquesta xarxa. En general Internet no és un lloc segur, i per tant és molt important que totes aquelles empreses o persones que treballen en xarxa, utilitzin aplicacions que els puguin oferir un grau important de seguretat, de forma que es minimitzin les possibles pèrdues o danys causats per usuaris no autoritzats. Això últim, es pot extrapolar a qualsevol usuari individual, fins i tot sense accés a cap xarxa.

L'eina més important per implementar aplicacions segures és la criptografia, una branca de les Matemàtiques que tracta sobre l'escriptura secreta.

Una xifra o criptosistema és un mètode secret d'escriptura, mitjançant el qual un text en clar es transforma en un text xifrat o criptograma. El procés de

transformar text en clar en text xifrat s'anomena xifratge; el procés invers, transformar text xifrat en text en clar, s'anomena desxifratge. Tant el xifratge com el desxifratge són controlats per una o més claus criptogràfiques.

La criptografia i una disciplina complementària anomenada criptoanàlisi es coneixen conjuntament amb el nom de criptologia. La criptografia s'ocupa del disseny de xifres. La criptoanàlisi s'ocupa de trencar xifres. La motivació del criptoanalista pot ser l'interès intrínsec de descobrir el text en clar xifrat i/o la clau emprada, o bé ser de caire cientificotècnic (verificació de la seguretat de la xifra). El vessant cientificotècnic de la criptoanàlisi és essencial per a la depuració de les xifres i és molt útil per al progrés de la criptografia.

2.2 Criptografia

Inicialment, la criptografia va aparèixer per a resoldre la necessitat de comunicar informació de forma segura en presència d'un adversari (normalment en un context militar o diplomàtic). Actualment, però, engloba molts altres problemes, com per exemple l'autenticació, el xifratge, la integritat de la informació, la distribució de claus, etc.

Aprofundint en aquests conceptes, podem veure que, en primer lloc, les aplicacions necessiten assegurar que els usuaris són realment usuaris autoritzats. La comprovació de la identitat en informàtica s'anomena **autenticació**. De la mateixa manera que un carnet d'identitat o un carnet de conduir són una forma d'autenticació, quan utilitzem un ordinador, generalment utilitzem un nom d'usuari i una contrasenya per autenticar-nos. La criptografia ofereix d'altres mètodes molt més segurs com per exemple la signatura digital o els certificats.

Les aplicacions informàtiques també necessiten protegir les dades d'accessos no autoritzats si aquests es produeixen, ja que les dades emmagatzemades en un disc poden ser robades o visualtizades. Així, la criptografia també ens ofereix solucions per tal de mantenir la **confidencialitat** (no volem que ningú pugui llegir les nostres dades) i la **integritat** de les dades (tampoc no volem que ningú no les pugui modificar).

És fàcil protegir les dades d'un disc dur xifrant els arxius de forma que un usuari no autoritzat no pugui visualitzar les dades tot i que en tingui accés a les mateixes.

De la mateixa manera, també cal assegurar la confidencialitat i la integritat de les dades enviades a través d'una xarxa, que poden rebre tota sèrie d'atacs i són especialment susceptibles de ser capturades.

Per exemple, és molt important pel comerç electrònic assegurar que les comunicacions són segures i ningú no podrà robar dades confidencials com ara el número de targeta de crèdit. Per aquesta raó la majoria dels navegadors de pàgines Web suporten SSL, un protocol criptogràfic que xifra la informació abans de que sigui transmesa a través de la xarxa Internet. SSL permet realitzar les compres a través d'Internet utilitzant les targetes de crèdit sense preocupar-se per l'estat final del compte corrent.

Un altre exemple important és el correu electrònic, ja que aquest és un mitjà molt fàcil de robar i de modificar. La Criptografia ajuda a fer que sigui molt difícil llegir els missatges xifrats i que el procés de modificació d'un correu electrònic sigui molt complicat afegint signatures digitals al correu.

Aprofundirem en els aspectes relacionats amb la confidencialitat de les dades, que és el tema clau al voltant del qual girarà l'aplicació desenvolupada en aquest treball final de carrera.

2.3 Confidencialitat

Una forma de protegir la informació dels ulls estranys és el xifratge, un procés pel qual un text en clar es transforma matemàticament en un text xifrat. Posteriorment, el desxifratge, transformarà un text xifrat en text en clar.

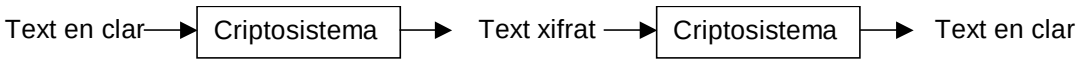


Figura 1. Funcionament bàsic d'un criptosistema

Per protegir totalment les dades del disc dur, aquestes s'haurien de xifrar abans d'escriure-les al disc, o bé assegurar la neteja del disc del text en clar una vegada xifrat l'arxiu. Cal tenir present que en l'actualitat no és suficient amb indicar que es vol esborrar un arxiu, ja que aquesta operació no esborra físicament les dades del disc dur, sinó que les marca com a esborrades. Per a esborrar les dades realment caldria sobreescriure els sectors ocupats per l'arxiu amb el text en clar.

Habitualment els xifradors fan ús d'una clau per xifrar i desxifrar les dades. Una clau és un valor secret, com per exemple una contrasenya. Les claus s'han de considerar com una seqüència de bits, i en el cas de l'aplicació desenvolupada s'obtenen a partir de la contrasenya introduïda per l'usuari. La clau és molt important en aquest procés, ja que xifrar el mateix text en clar utilitzant diferents claus tindrà com a resultat diferents texts xifrats. De la mateixa manera, el text xifrat només es pot desxifrar utilitzant la clau apropiada.

2.4 Criptosistema de clau compartida (simètric)

Els criptosistemes de clau compartida són aquells en els quals l'emissor i el receptor comparteixen una mateixa clau per a xifrar i desxifrar missatges. En el nostre cas l'emissor i el receptor podrien ser la mateixa persona si ens limitem a protegir les dades del nostre disc dur sense enviar-les enlloc.

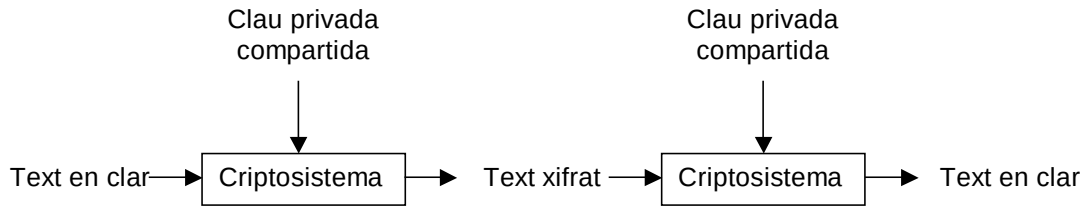


Figura 2. Funcionament d'un criptosistema de clau compartida

Els criptosistemes de clau compartida presenten un dilema, derivat del fet que la clau ha de ser coneguda per l'emissor i pel receptor i a més ha de ser secreta. Això implica que en algun moment, l'emissor i el receptor han d'haver intercanviat la clau de forma segura. Aquest problema, no el tindrem en el cas que l'emissor i el receptor siguin la mateixa persona.

A més cal tenir present que si aquesta clau es compartida per més d'una persona, si la clau és descoberta, es posa en perill la seguretat del sistema format per tot el conjunt de persones. En el nostre cas, i per tal d'evitar aquest problema és suficient en tenir la precaució de no escriure la contrasenya utilitzada en cap lloc accessible, i a més tenir la precaució d'utilitzar un joc de contrasenyes en comptes d'una única contrasenya.

2.4.1 Xifres de flux

Per obtenir un criptosistema segur es necessita tenir el mateix nombre de bits de clau com de bits de text per a xifrar. Ara bé, disposar d'una clau amb el mateix nombre de bits que el text a xifrar comporta una llargària de clau molt gran, que seria molt difícil de manejar i mantenir en secret. A més, es donaria la paradoxa que es necessitaria un canal segur per poder enviar la clau, i si aquesta té la mateixa longitud que el text, res no impedeix enviar directament el text per aquest canal segur.

Per solucionar aquest fet, les xifres de flux sorgeixen com una aproximació optimitzada del xifratge de Vernam. La idea és utilitzar un generador de claus pseudoaleatori, que permeti la generació d'una clau prou llarga a partir d'una

clau inicial més curta, la llavor. Aquesta llavor és la que s'obté a partir del tractament informàtic d'una contrasenya introduïda per l'usuari.

La seqüència no pot ser totalment aleatòria ja que hem de poder xifrar i desxifrar els fitxers i, per tant, donada una llavor, aquesta ha de generar la mateixa seqüència com a clau, és a dir els algorismes de generació han de ser deterministes, la qual cosa fa que es perdi en seguretat respecte a la idea original del xifratge de Vernam.

Convé no oblidar que els criptosistemes de clau compartida basen la seguretat en el fet que la clau emprada per a xifrar i desxifrar només la coneixen l'emissor i el receptor. En el xifratge de flux, si bé la clau no es fa servir directament per a xifrar, convé igualment que no es faci pública, ja que l'algorisme determinista és conegut i es podria obtenir la seqüència de xifratge a partir d'aquest i de la clau. També convé assegurar que la seqüència obtinguda compleix una sèrie de requisits que l'aproximin a una seqüència totalment aleatòria.

El funcionament d'un criptosistema de flux es pot veure a la següent figura:

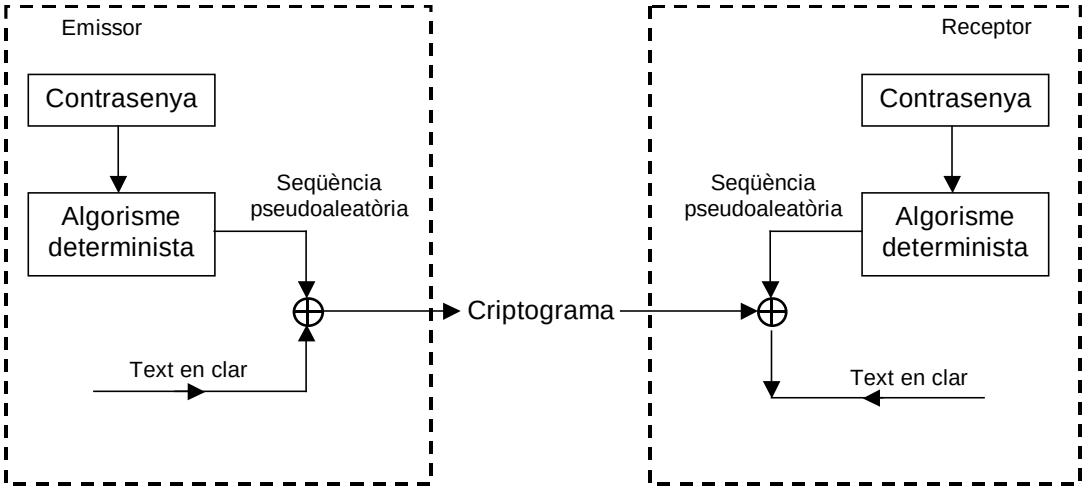


Figura 3. Procediment de xifrat de flux

Per a xifrar el missatge, l'emissor va sumant cada bit del missatge en clar, amb cada bit de la seqüència de xifratge. Quan el receptor rep el missatge xifrat fa servir el mateix algorisme determinista i la clau privada, que comparteix amb l'emissor, per a obtenir la mateixa seqüència de xifratge. Així, sumant bit a bit el missatge que li arriba (criptograma) amb la seqüència resultant de l'algorisme el receptor obté el text en clar enviat per l'emissor.

Posteriorment, en parlar dels aspectes concrets de la implementació es farà menció als algorismes deterministes que s'utilitzen habitualment i quina ha estat la solució adoptada.

2.4.2 Xifres de bloc

Es denomina xifratge en bloc aquell en el que es xifra el missatge original agrupant els símbols en blocs de dos o més elements.

Els esquemes de xifratge de bloc presenten les següents propietats:

- Cada símbol es xifra de manera dependent dels adjacents.

- Cada bloc es xifra sempre de la mateixa manera, independentment de la seva posició en el missatge.

- Dos missatges originals iguals, xifrats amb la mateixa clau, produeixen sempre missatges xifrats iguals.

- Un missatge es pot desxifrar parcialment, a partir del bloc que interessi.

Tots els algorismes de xifratge en bloc tenen bàsicament la mateixa estructura de funcionament. Així, consten de:

- Transformació d'entrada. Aquesta transformació introdueix aleatorietat o dificulta els atacs per anàlisis lineal o diferencial com per exemple a l'algorisme DES.

- Nombre determinat d'iteracions d'una certa funció f, no lineal, que combina els elements que formen part del bloc de text en clar amb els elements que formen part de la clau.

- Algorisme d'expansió de clau. Generalment la clau d'usuari (curta), K, s'utilitza per a generar un seguit de subclaus, K_i, a partir d'una certa funció f_K prou complicada, i són aquestes subclaus les que actuen en cada iteració.

- Transformació de sortida que serveix per a que les operacions de xifratge i desxifratge siguin simètriques (desfan el canvis introduïts a la transformació d'entrada).

Per tant, a partir d'un bloc de text en clar d'una longitud fixada, i una clau, aquests tipus d'algorismes executen determinades operacions més o menys complicades fins a obtenir el text xifrat corresponent. A més, les operacions

anteriors han de permetre que es pugui tornar a obtenir el mateix text en clar a partir del text xifrat i la clau, si es porta a terme el procés de desxifratge.

2.4.2.1 Modes de funcionament[1]

Les xifres de bloc presenten diferents modes de funcionament que determinen la forma com s'encadenen els diferents blocs en els que es divideix el text en clar. Bàsicament en parlarem dels modes ECB i CBC. El primer perquè descriu el funcionament bàsic i el segon que ens interessarà especialment ja que és el que utilitzaran els algorismes implementats en la nostra aplicació.

- ECB

El mode bàsic de funcionament d'un criptosistema de bloc és el que es coneix amb la sigla ECB (Electronic Code Book). Com que la majoria de vegades el text que s'ha de xifrar té una llargada superior a la longitud del bloc amb el qual treballa el criptosistema, el que es fa és partir el text que cal xifrar, M, en diversos blocs, M1, M2, ..., cada un dels quals té la llargada corresponent al bloc per a xifrar. D'aquesta manera es xifra cada un dels blocs amb una mateixa clau, K, per a obtenir el text xifrat resultant: Y = Y1Y2 ...

El mode ECB presenta inconvenients respecte a la seguretat, ja que el fet que el xifratge de dos blocs sigui totalment independent fa que sigui vulnerable a determinats atacs que no fan aconsellable la seva utilització per xifrar textos llargs, encara que si és un bon mode per codificar missatges curts com ara identificadors o contrasenyes on no es supera la llargada del bloc.

- CBC

El CBC (Cipher Block Chaining) consisteix en l'encadenament dels blocs per al xifratge, de manera que es creï una dependència del xifratge de cada bloc amb el que el precedeix, tal com mostra la figura següent.

Altres modes de funcionament dels criptosistemes de bloc que ofereixen diferents esquemes són el mode CFB (Cipher Feedback) i el mode OFB (Output Feedback).

[1] Es poden consultar a: http://csrc.nist.gov/publications/fips/fips81/fips81.htm

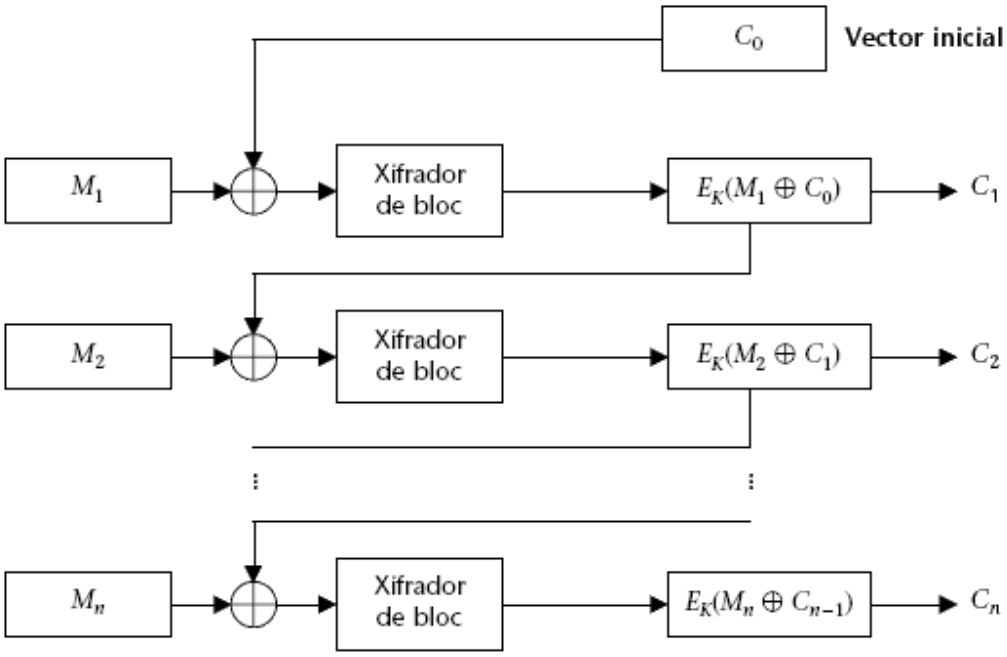


Figura 4. Esquema de funcionament del mode CBC

Com es pot observar, abans de xifrar un bloc de text en clar, es combina amb el bloc xifrat anteriorment mitjançant una funció XOR bit a bit. D'aquesta manera cada bloc depèn de tots els anteriors. Per desxifrar el text clar d'un bloc xifrat no hi ha prou en conèixer la clau, sinó que també es necessita el bloc anterior.

El primer bloc no disposa de cap bloc anterior, per aquesta raó s'utilitza un vector inicial (IV) aleatori de la mateixa llargada que els blocs.

2.4.2.2 L'estàndard DES (Data Encryption Standard)

El criptosistema DES és un criptosistema de xifratge de bloc que xifra blocs de dades de 64 bits de llargada per mitjà d'una clau de 56 bits.

L'any 1973, el National Bureau of Standards (NBS), va organitzar un concurs sol·licitant un "algorisme de xifratge per a la protecció de les dades d'ordinador durant la seva transmissió i emmagatzematge". Curiosament, aquesta sol·licitud s'apropa molt a la base del nostre objectiu, que ens serveix per justificar la seva utilització, encara que afortunadament nosaltres no haurem de dissenyar ni implementar l'algorisme.

Els requisits d'aquest algorisme eren que la seva implementació electrònica havia de ser fàcil i barata, que proporcionés un alt nivell de seguretat i que

aquesta seguretat depengués només del secret de la clau i no del secret de l'algorisme, ja que aquest seria públic.

L'any 1974, IBM presentà una proposta basada en el seu sistema propietari LUCIFER. Aquest sistema rep propostes de canvi per part de l'Agència Estatal de Seguretat (NSA) dels Estats Units (longitud de la clau i disseny de les caixes) que són acceptades. Aquestes modificacions donen lloc al DES.

Finalment l'any 1977, s'accepta el DES com el criptosistema estàndard a utilitzar per a protegir qualsevol tipus de dades electròniques.

Actualment el DES ha estat substituït per l'AES (Advanced Encryption System) com a estàndard, però encara hi ha moltes aplicacions que encara el continuen fent servir.

Cal tenir present que les modificacions introduïdes per la NSA han resultat les dues febleses més greus del criptosistema DES.

Així la poca llargada de la clau, de només 56 bits limita molt l'univers de claus possibles (2^{56} = 72.057.594.037.927.936 claus ≈ 72.000 bilions de claus), fa possible avui dia un atac per força bruta si es disposa de la suficient potència de càlcul.

L'altre modificació objecte de controvèrsia va ser l'arbitrarietat de les caixes S, que podria obeir a l'existència d'una clau mestra que permetria a la NSA desxifrar qualsevol missatge sense tenir-ne la clau original.

Descripció del funcionament[2].

L'algorisme DES ha estat dissenyat per a xifrar i desxifrar blocs de dades de 64 bits utilitzant una clau DES de 64 bits. Aquesta clau consisteix en 64 digits binaris (0 i 1) dels quals 56 han estat generats de forma aleatòria i s'utilitzen directament en l'algorisme. Els altres 8 bits, es poden utilitzar per a la detecció d'errors com a bits de paritat.

El funcionament de l'algorisme DES queda determinat a la figura de la pàgina següent. Com podem observar, se subministra a l'algorisme un bloc d'entrada, M, sobre el qual s'aplica una permutació inicial, σ, d'on s'obté $T_0 = \sigma(M)$. Aquest bloc es separa en dues parts de 32 bits cadascuna E (esquerra - conté els 32 bits més significatius) i D (dreta - conté els 16 bits menys significatius). Després es sotmet aquest nou bloc a 16 iteracions d'una funció complexa que depèn de la clau. Cadascuna d'aquestes iteracions consisteix en la suma mòdul 2 (XOR) de la part esquerra amb una transformació de la part dreta a través d'una funció f

[2] Es pot consultar a: http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

que depèn de la clau. Posteriorment s'intercanvien la part esquerra i la part dreta, amb excepció de la última iteració. Finalment el resultat es transposa per mitjà de la permutació de sortida σ^{-1}, fet que permet que es pugui aplicar el mateix esquema pel procés de desxifratge.

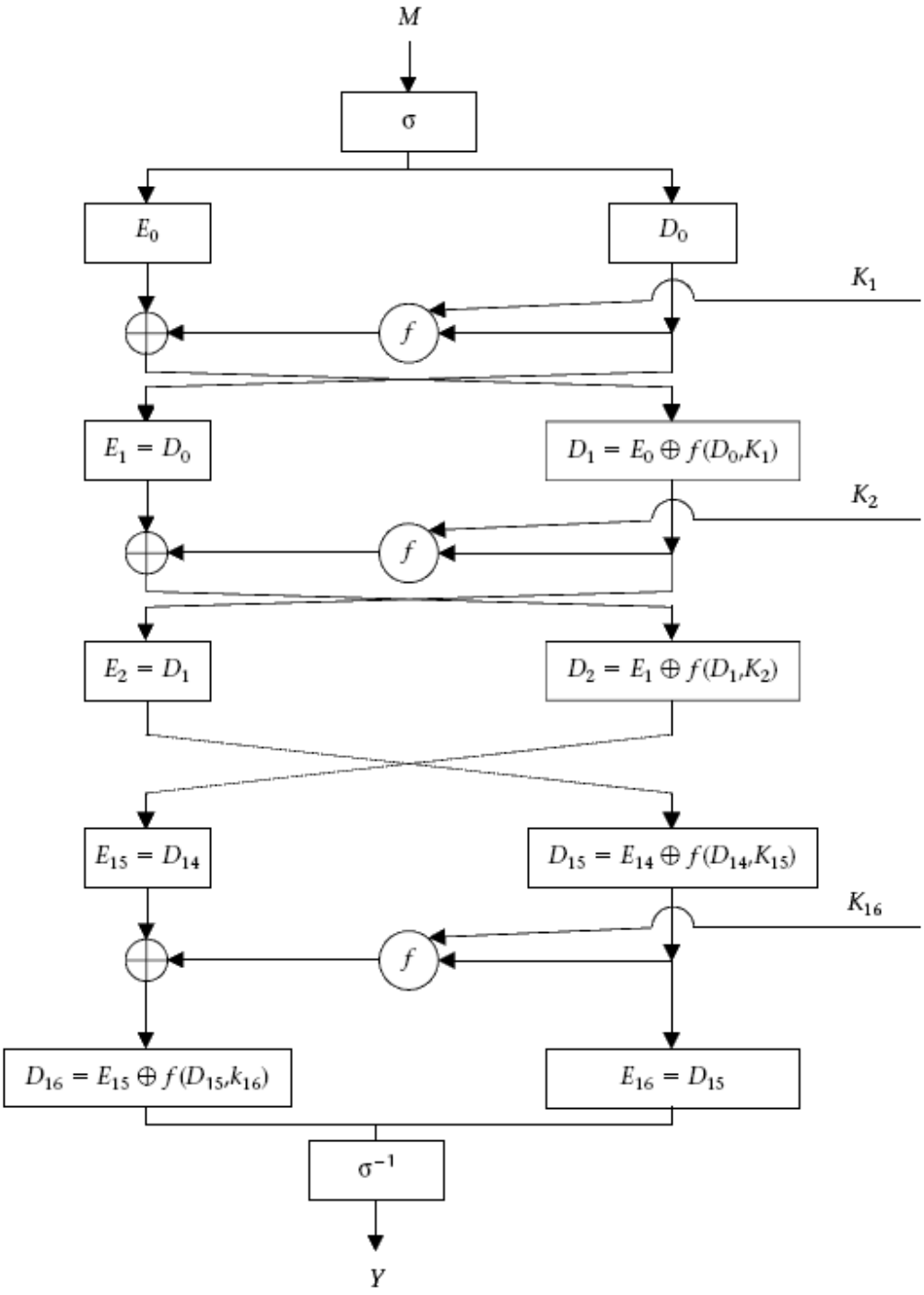


Figura 5. Esquema general de funcionament del DES

La funció complexa f, és un conjunt d'operacions que es combinen com es pot veure a la figura següent.

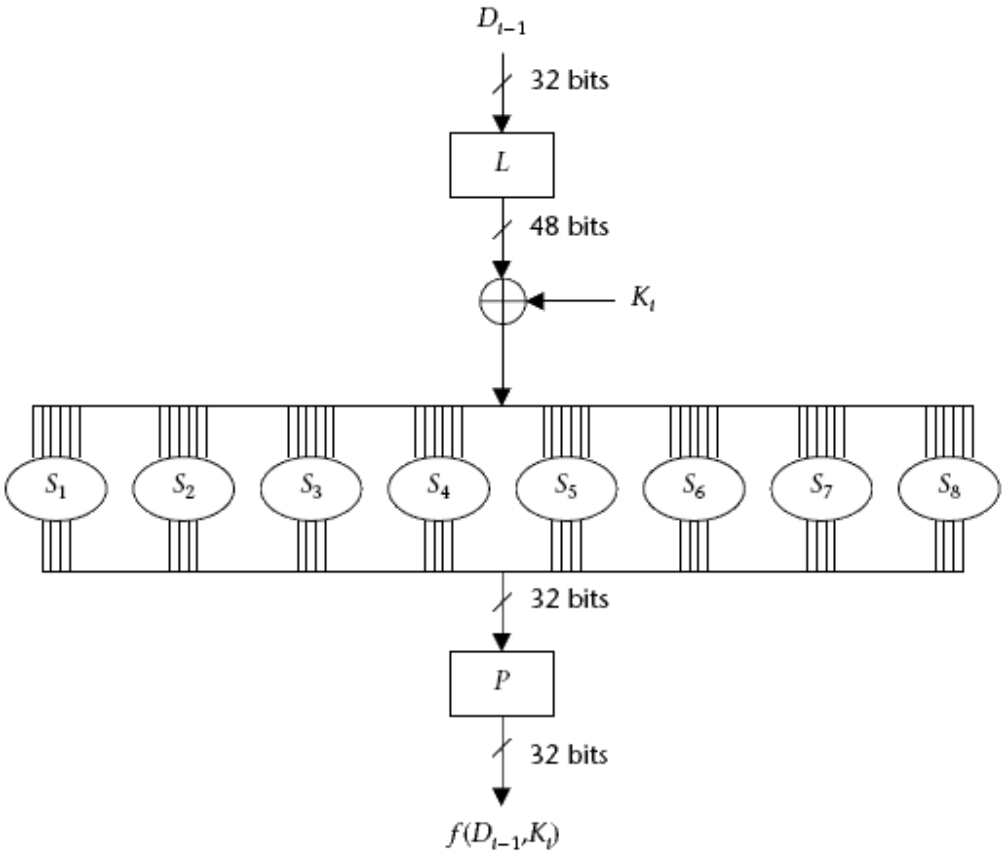


Figura 6. Descripció de la funció f

En primer lloc es construeix un vector de 48 bits a partir dels 32 bits d'entrada (D_{i-1}) mitjançant una expansió lineal (L), que reordena i repeteix alguns dels bits d'entrada.

A continuació, es fa una suma bit a bit de la meitat expandida, $L(D_{i-1})$, amb la clau Ki. El resultat d'aquesta operació es descomposa en vuit blocs de 6 bits cada un.

Cada bloc, B_j, s'usa com a entrada del que s'anomena una caixa S. Aquestes caixes són les responsables de la no linealitat del DES. Cada caixa, S_j, rep el seu bloc corresponent de 6 bits B_j i en retorna un de 4 bits de llargada, d'acord amb unes taules prefixades.

Cada caixa està formada per una taula de 4 files i 16 columnes . Cada cel·la d'aquesta taula conté un nombre enter entre 0 i 15. Els principis d'elecció dels continguts d'aquestes caixes és informació classificada pel govern dels Estats Units i una de les grans controvèrsies d'aquest algorisme.

El criteri per mitjà del qual s'assigna una fila i una columna d'una caixa a Bj és el següent: l'índex j fixa la caixa Sj, l'enter corresponent a b1 b6 selecciona la fila i l'enter que correspon a b2 b3 b4 b5 determina la columna.

A continuació, es prenen els blocs resultants de les caixes S i es concatenen fins a obtenir un bloc de 32 bits de llargada. Finalment, s'aplica sobre aquest darrer bloc una permutació P per tal d'obtenir el valor $f(D_{i-1},K_i)$.

Generació de les subclaus.

DES utilitza 16 subclaus K_i de 48 bits obtingudes a partir de la clau inicial de 56 bits. L'algorisme de generació d'aquestes subclaus s'il·lustra a la següent figura.

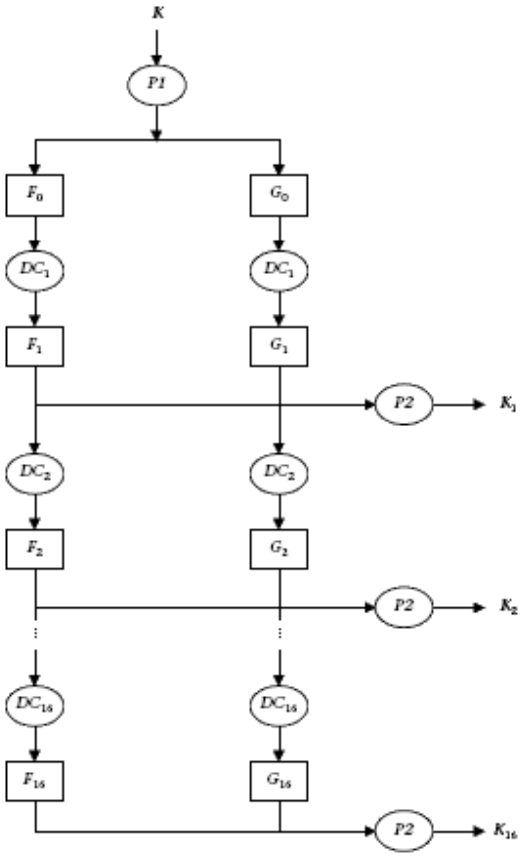


Figura 7. Esquema de generació de subclaus

En primer lloc, es fa passar la clau inicial, K, de 56 bits per una permutació inicial P1. El resultat de fer la permutació P1(K) és divideix K en dues meitats, F_0 i G_0, de 28 bits cadascuna. Els blocs F_0 i G_0 es desplacen cap a l'esquerra per a obtenir cada subclau K_i.

Si representem per F_i i G_i els valors emprats per a obtenir K_i, tenim, $F_i=DC_i(F_{i-1})$, $G_i = DC_i (G_{i-1})$, en què DC_i és un desplaçament circular cap a l'esquerra de 1 o 2 posicions en funció del valor de i.

Finalment, la subclau K_i s'obté després d'aplicar una segona permutació P2 al resultat de concatenar els blocs F_i i G_i.

Comentaris a l'algorisme.

Des de l'aparició del DES ja van sorgir diferents veus crítiques sobre aquest criptosistema. Les dues febleses més greus que se li atribueixen són la poca llargada de la clau, de només 56 bits, i l'arbitrarietat de les caixes S, que podria obeir a l'existència d'una clau mestra que permetés desxifrar qualsevol missatge sense tenir-ne la clau original.

Actualment, es considera que una llargada de la clau de 56 bits és insegura, ja que es pot trencar la xifra amb poques hores, això si, utilitzant supercomputadors dissenyats per aquesta funció. Ara bé, per una aplicació com la nostra, DES pot continuar sent totalment vàlid, ja que pot ser suficient per amagar la informació als ulls de la majoria de les persones i a la vegada presenta una gran velocitat de processament.

Per a resoldre el problema de la llargada de la clau i continuar usant el DES com a criptosistema segur, s'utilitza el xifratge triple, que en el cas del DES és conegut com a triple DES o DESede.

2.4.2.3 Triple DES (Data Encryption Standard)[3]

DESede, també anomenat triple DES, és una variant de l'algorisme DES. De fet, existeix diferents variants i implementacions de triple DES. Les dues versions més utilitzades difereixen en la longitud de la clau utilitzada.

En la versió més segura de l'algorisme els blocs de text en clar es transformen en text xifrat utilitzant tres claus DES, i tres aplicacions de l'algorisme DES. Cadascuna d'aquestes aplicacions utilitza una clau diferent, i segueix els següents passos:

[3] Es pot consultar a http://csrc.nist.gov/publications/nistpubs/800-67/SP800-67.pdf

1. El text en clar és xifrat utilitzant la primera clau.
2. El resultat del primer pas és desxifrat utilitzant la segona clau.
3. El resulttat del segon pas és xifrat utilitzant la tercera clau, produint el text xifrat final.

Si aquest és el procés de xifratge, el procés de desxifratge és el següent:

1. El text xifrat és desxifrat utilitzant la tercera clau.
2. El resultat del primer pas és xifrat utilitzant la segona clau.
3. El resultat del segon pas és desxifrat utilitzant la primera clau, produint el text en clar original.

L'algorisme DESede és molt més difícil de criptoanalitzar que DES, ja que la longitud de la clau s'incrementa fins als 168 bits com a conseqüència de la combinació de les tres claus DES de 56 bits.

Una variant menys segura d'aquest algorisme utilitza dos claus DES en comptes de tres (112 bits de clau). En aquest cas, la primera clau també fa el paper de la tercera.

2.5 Criptosistema de clau pública (asimètric)

El concepte de criptosistema de clau pública ha suposat una revolució en la criptografia moderna. Aquest concepte, que permet superar els inconvenients dels criptosistemes de clau privada, va ser proposat per W. Diffie i M.E. Hellman en l'article "New directions in cryptography" aparegut l'any 1976. La idea, que es pot titllar de revolucionària, era permetre un intercanvi segur de missatges entre emissor i receptor sense ni tan sols haver-se de trobar prèviament per acordar una clau secreta comuna.

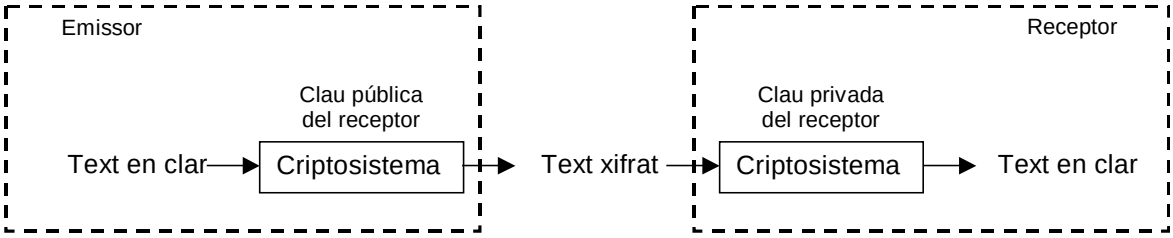


Figura 8. Funcionament d'un criptosistema de clau pública

Un criptosistema de clau pública s'adreça més sovint a una xarxa d'usuaris que no a una sola parella. En aquest criptosistema cada usuari, u, té associada una

parella de claus < P_u,S_u >: la clau pública, P_u, que es publica amb el nom de l'usuari en un directori públic que tothom pot llegir, mentre que la clau privada, S_u, només la coneix u. Els parells de claus es generen mitjançant un algorisme de generació de claus.

Per a enviar un missatge secret m a u, tothom de la xarxa fa servir el mateix mètode:

1) Cercar P_u.

2) Calcular c = E(P_u,m), en què E és un algorisme públic de xifratge.

3) Enviar c a l'usuari u.

En rebre el text xifrat c, l'usuari u el pot desxifrar de la manera següent:

1) Cercar la seva clau privada S_u.

2) Calcular D(S_u,c), en què D és un algorisme públic de desxifratge.

Perquè aquest procediment funcioni cal que D(S_u,E(P_u,m)) = m. D'aquesta manera la gestió de claus queda simplificada, perquè ara només hi ha n parelles de claus per a n usuaris, en comptes de les n(n - 1) / 2 claus que calien amb la criptografia de clau compartida.

En el nostre cas la implementació de l'aplicació no utilitzarà aquesta tècnica, sinó que es basarà en els criptosistemes de clau compartida amb xifratge de bloc.

3 Disseny del programa

3.1 Esboç del disseny

Com ja he comentat anteriorment, dissenyar una aplicació de xifratge implica fer un balanç a tres bandes entre el cost de "perdre" les dades per un atac informàtic, el cost efectiu de desenvolupar una aplicació segura i els canvis en la facilitat d'ús de la nova aplicació. Seguint aquest concepte, el disseny del programa es basa en dos conceptes, d'una banda simplificar la instal·lació de l'aplicació a l'usuari final i proporcionar una interfície gràfica que en faciliti l'ús de l'usuari final i per altra banda, racionalitzar el cost d'implementació de l'aplicació sempre tenint present que es tracta del desenvolupament d'un treball universitari.

L'aplicació consisteix en una utilitat que treballa a nivell de confidencialitat de les dades, és a dir, permet xifrar i desxifrar fitxers a partir d'una contrasenya introduïda per l'usuari. Aquesta utilitat disposa d'una interfície gràfica que permet a l'usuari interactuar amb l'aplicació.

Quan l'usuari accedeix a l'aplicació, aquesta mostra un arbre de directoris a la pantalla on es pot seleccionar un fitxer o carpeta sobre el qual es poden realitzar les operacions de xifratge o desxifratge.

Una vegada escollit el fitxer o carpeta, l'usuari ha de seleccionar l'operació que vol realitzar amb l'arxiu (xifratge/desxifratge), l'algorisme que s'utilitzarà per a realitzar l'operació i el nom que tindrà l'arxiu de sortida.

Després d'escollir el fitxer i les opcions, l'usuari ha d'introduir una contrasenya que servirà per xifrar o desxifrar l'arxiu. L'aplicació comprova que aquesta contrasenya sigui prou llarga per a generar la clau privada.

Si l'usuari ha escollit una carpeta, l'aplicació crea un arxiu comprimit (de tipus zip) amb el contingut de la carpeta. Una vegada comprimida la carpeta, l'aplicació treballa de la mateixa manera que si d'un únic arxiu es tractés.

En qualsevol cas, l'aplicació permet a l'usuari que decideixi que s'ha de realitzar amb els fitxers originals, conservar-los o esborrar-los.

Finalment, l'aplicació realitza l'acció escollida per l'usuari i mostra una sèrie de missatges indicant el progrés de les diferents operacions (validació de les dades, existència dels arxius, xifrar, ...).

El programa ha estat desenvolupat en el llenguatge de programació Java, utilitzant la versió 1.5.0 del Java Development Kit. Per a la interfície gràfica d'usuari s'ha utilitzat els components de JFC/Swing. Per a les operacions de

xifratge i desxifratge, s'ha utilitzat la llibreria criptogràfica de Java (Java Cryptography Extension JCE) per tal de facilitar al màxim la instal·lació de l'aplicació, en comptes d'utilitzar llibreries d'altres fabricants com ara Cryptix o IAIK, tot i ser gratuïtes i/o de lliure distribució. Les llibreries de la JCE ja estan integrades a la versió de Java utilitzada i per tant no cal descarregar cap component addicional.

3.2 Planificació temporal

L'apartat del pla docent corresponent a la temàtica d'aquest TFC, indicava una sèrie de tasques a realitzar per al correcte desenvolupament de la feina.

La descomposició inicial de les tasques era:

1. Cerca d'informació sobre el xifratge simètric de flux
2. Cerca d'informació sobre la generació de pseudoaleatoris a partir de llavors
3. Implementar el sistema de xifratge desxifratge
4. Afegir-hi el mòdul xifrador/desxifrador
5. Si s'escau, afegir l'opcionalitat de poder xifrar/desxifrar carpetes
6. Elaboració de la memòria: objectiu del projecte, estat de l'art i recursos
7. Elaboració de la memòria: disseny del programa (amb diagrames senzills i descripció a alt nivell)
8. Elaboració de la memòria: manual d'usuari, jocs de proves, incidències i conclusions
9. Realitzar la presentació del projecte amb PowerPoint o altres aplicacions

A partir d'aquest llistat i afegint aquelles tasques que es van creure necessàries per complementar les ja donades, es va redactar una planificació temporal, intentant que fos realista i permetés el desenvolupament del TFC de forma pausada i repartida al llarg de tot el trimestre.

A continuació es presenta la planificació temporal inicial.

Dates	**Tasca general**	**Tasca específica**
19/9 – 26/9	Pla de Treball	Reflexió sobre la temàtica del treball escollit Recerca bàsica d'informació
25/9 – 26/9	Pla de Treball	Redacció del pla de treball

26/9 – 2/10	Documentació	Cerca d'informació sobre el xifratge simètric de flux
3/10 – 9/10	Documentació	Cerca d'informació sobre el xifratge simètric de blocs
10/10 – 16/10	Documentació	Cerca d'informació sobre la generació de pseudo-aleatoris a partir de llavors
17/10 – 23/10	Disseny	Disseny del sistema de xifratge/desxifratge a implementar
21/10 – 24/10	PAC 2	Resolució i lliurament de la PAC 2
24/10 – 20/11	Programació	Implementació del sistema de xifratge/desxifratge
14/11 – 20/11	Disseny	Disseny de la interfície gràfica d'usuari
18/11 – 21/11	PAC 3	Resolució i lliurament de la PAC 3
21/11 – 4/12	Programació	Implementació de la interfície gràfica d'usuari
5/12 – 11/12	Memòria	Redacció dels objectius del projecte, l'estat de l'art i els recursos utilitzats
17/12 – 18/12	Memòria	Redacció del manual d'usuari
22/12 – 24/12	Programació	Proves d'integració del programari
22/12 – 24/12	Memòria	Redacció del disseny final del programa
27/12 – 31/12	Memòria	Descripció dels jocs de proves utilitzats, les incidències detectades i redacció de conclusions.
2/01 – 3/01	Presentació	Realitzar la presentació del projecte

3/01	Memòria	Lliurament de la prememòria
3/01 – 10/01	Memòria	Correcció de la prememòria i de la presentació. Detalls finals.
10/01	Memòria	Lliurament Final

Per diferents motius de caràcter laboral ha estat impossible complir amb els terminis marcats inicialment i a partir de la segona setmana del mes de novembre la feina s'ha endarrerit molt, tot i que al final s'ha aconseguit enllestir el projecte d'acord amb les especificacions inicials.

4 Aspectes concrets de la implementació

La implementació de l'aplicació es pot dividir en tres fases diferents, la primera i més important, que ocupa el problema del xifratge/desxifratge.

La segona i no menys important fa referència a la interfície gràfica d'usuari que permet la interacció aplicació-usuari d'una forma fàcil.

Per últim, la tercera correspon al tractament del xifratge de carpetes, una tasca opcional que també s'ha afegit a l'aplicació.

4.1 Xifratge

4.1.1 Generació de la clau

Ja hem comentat anteriorment, la necessitat de tenir el mateix nombre de bits de clau com de bits de text per a xifrar si es vol que el criptosistema sigui segur, cosa força impossible en una aplicació com la nostra en la qual la contrasenya té pocs bits en comparació amb el text a xifrar.

Per solucionar aquest fet, la primera idea va ser utilitzar un generador de claus pseudoaleatori, que permetés la generació d'una clau prou llarga a partir d'una llavor (clau més curta) obtinguda de la contrasenya introduïda per l'usuari.

Per poder obtenir una seqüència pseudoaleatoria es poden fer servir generadors lineals (generadors congruents, registres de desplaçament linealment realimentats LFSR) o generadors no lineals (registres de desplaçament realimentats no linealment NLFSR, generador de Geffe, generador de Beth-Piper, generador de Massey-Rueppel).

Des de bon principi, s'ha descartat la utilització de generadors lineals ja que, d'una banda, els generadors congruents no són prou segurs i, d'altra banda, els generadors lineals són fàcilment predictibles degut a que tenen una baixa complexitat lineal en comparació amb el període.

Dels generadors no lineals, els NLFSR tenen funcions de realimentació difícils d'especificar i que a la vegada presentin períodes llargs. Per tant s'acostuma a treballar amb combinadors no lineals, com ara el generador de Geffe, de Beth-Piper o de Massey-Ruepel, que basen el seu funcionament en la combinació de diversos LFSR.

Aquests generadors no lineals presenten característiques semblants des del punt de vista d'implementació, ja que es combinen diferents LFSR de forma que s'obtenen complexitats lineals superiors a les que s'obtenien amb els generadors lineals.

D'aquests tres tipus de generadors en principi s'havia optat per utilitzar un generador de Geffe, donat que ja es coneixia la seva implementació a través de les pràctiques realitzades a l'assignatura de Criptografia.

Aquesta implementació es pot consultar a la carpeta Annexos\Annex2 on hi ha un document explicatiu de la seva implementació així com els arxius font i binaris de les classes necessàries per comprovar el seu funcionament executant la classe Geffe. Execució: java Geffe [Grau_dels_LFSR] [longitud_sortida_bytes]

Tot i ser un bon punt de partida, finalment aquesta via es va descartar, ja que la idea inicial era fer una aplicació d'acord amb l'estàndard, i consultant la documentació sobre el tema, es va comprovar que ja hi havia publicat un estàndard per desenvolupar aplicacions de xifratge basades en contrasenya[4]. Aquest estàndard anava més enllà de la generació de la clau i indicava tot el procés a seguir per realitzar els processos de xifratge i desxifratge, introduint algunes variacions interessants que dificulten els atacs de diccionari.

A més, el fet d'utilitzar la implementació dels algorismes proporcionada pels proveïdors, simplifica molt la programació i estandarditza el codi, de forma que sigui molt més fàcil el manteniment i l'ampliació amb nous algorismes més segurs, així com eliminar aquells que puguin quedar obsolets amb el pas del temps.

4.1.2 L'estàndard PBE (Sal i vinagre)

En moltes aplicacions de criptografia de clau pública, la seguretat depèn del valor secret d'una contrasenya. Això implica un perill ja que habitualment les contrasenyes tenen poca llargària, o es desen en llocs poc segurs, cosa que les fa atractives a atacs per diccionari i compromet per tant la seguretat.

D'altra banda, la contrasenya no es pot aplicar directament com una clau del criptosistema sinó que cal processar-la informàticament per adaptar-la a l'algorisme escollit. Això permet utilitzar una tècnica per protegir aquestes aplicacions d'atacs per diccionari. Aquesta tècnica consisteix en combinar la contrasenya amb un conjunt de bytes ("*salt*" - sal) per obtenir la clau.

[4] RSA Laboratories. PKCS#5: Password Based Encryption Standard. Versió 2.0, Març 1999.

D'aquesta manera s'aconsegueix incrementar el nombre de combinacions diferents que cal provar per a realitzar un atac per diccionari. Així, si per exemple s'utilitza un bit (0 o 1) per a la sal, es multiplica per 2 el número d'entrades al diccionari. L'estàndard definit al PKCS#5 indica un valor de 8 bytes per a la sal, i per tant les entrades en el diccionari es multipliquen per 2^{64} , un valor prou gran per desanimar els possibles atacants.

Una altra manera d'evitar atacs per diccionari és incrementar el cost de cerca de les claus per part dels atacants (posar-hi vinagre). Una forma fàcil de fer-ho és aplicar la funció a partir de la qual es deriven les claus més d'una vegada. El nombre recomanat d'iteracions és com a mínim de 1000. Cal tenir present, que el cost no és gaire important per l'usuari, ja que només ha de realitzar aquest procés una vegada, i en canvi l'atacant ho ha de fer amb cada una de les entrades del diccionari.

Sal i vinagre, formen la base del xifratge basat en contrasenya (PBE) i definits en el document PKCS#5 v1.5 com una funció de una contrasenya, un vector sal afegit i un nombre d'iteracions, on les dues últimes quantitats no cal mantenir-les necessàriament en secret.

4.1.2.1 El procés de xifratge

Els passos a realitzar per xifrar un text en clar a partir d'una contrasenya seguint aquest esquema són els següents:

1. Escollir un vector de 8 bytes com a sal i un valor pel nombre d'iteracions.

2. Aplicar la funció per obtenir la clau derivada al conjunt contrasenya, sal, iteració. S'obté una clau derivada amb 16 bytes.

3. Separar els 16 bytes de la clau derivada en una clau privada amb els 8 primers bytes i un vector d'inicialització (IV) amb els 8 últims.

4. Concatenar el text en clar amb una sèrie de bytes ("padding") de forma que la longitud total del text així format sigui múltiple de 8 bytes (mida del bloc). El valor dels bytes del farcit depèn del nombre de bytes que calgui afegir. Així si cal afegir 2 bytes, aquests tindran el valor 2, si cal afegir-ne 5 llavors valdran 5. D'aquesta manera es podrà recuperar el text original sense cap mena d'ambigüitat.

5. Xifrar el missatge amb l'algorisme de bloc (DES o RC2) en mode CBC amb la clau i el vector inicialització obtinguts al pas 3.

6. Donar com a sortida el text xifrat.

4.1.2.2 El procés de desxifratge

Els passos a realitzar per recuperar un text en clar a partir d'un text xifrat amb una contrasenya seguint aquest esquema són els següents:

1. Obtenir la sal (vector de 8 bytes) i el nombre d'iteracions.

2. Aplicar la funció per obtenir la clau derivada al conjunt contrasenya, sal, iteració. S'obté una clau derivada amb 16 bytes.

3. Separar els 16 bytes de la clau derivada en una clau privada amb els 8 primers bytes i un vector d'inicialització (IV) amb els 8 últims.

4. Desxifrar el missatge amb l'algorisme de bloc (DES o RC2) en mode CBC amb la clau i el vector inicialització obtinguts al pas 3. Si la longitud en bytes del text xifrat no és un múltiple de 8, donar un missatge d'error i finalitzar.

5. Separar la cadena de text que conté el "padding" del missatge obtingut, on aquesta cadena de text consisteix en un determinat nombre de bytes psLen cadascun d'ells amb el valor psLen. Si no és possible separar aquest missatge, donar un missatge d'error i finalitzar.

6. Donar com a sortida el text en clar.

4.1.3 Algorismes utilitzats

Una vegada presa la decisió d'utilitzar l'esquema definit anteriorment, calia veure quins eren els algorismes proporcionats per Sun a les seves llibreries que utilitzessin aquest estàndard. Actualment la versió 5 de Java (1.5.0) proporciona diferents algorismes basats en aquest esquema, que funcionen amb els algorismes DES, Triple DES i RC2. Dels tres algorismes, l'últim es va descartar per ser el que oferia menys seguretat, i finalment es va optar per implementar els esquemes PBEWithMD5AndDES i PBEWithSHA1AndDESede.

L'ús de l'esquema PBEWithMD5AndDES va ser definit a: RSA Laboratories, *"PKCS #5: Password-Based Encryption Standard",* versió 1.5, Novembre 1993.

L'ús de l'esquema PBEWithSHA1AndDESede va ser definit anys més tard a: RSA Laboratories, *"PKCS #5: Password-Based Cryptography Standard",* versió 2.0, Març 1999.

PBE (*passphrase-based encryption*) s'utilitza per obtenir una clau a partir d'una contrasenya i poder realitzar el xifrat/desxifrat amb aquesta clau. En la primera variant, s'utilitza MD5 per obtenir un resum de la contrasenya i, posteriorment aquest valor es utilitzat com a clau a l'algorisme DES. En la segona, s'utilitzen SHA1 i triple DES respectivament.

En els dos casos, per aconseguir el vector amb la sal, el procés ha estat obtenir els 8 primers bytes del resum generat amb la funció MD5 aplicats al resultat de concatenar la contrasenya i el fitxer amb el text en clar. Aquesta manera de generar la clau, ens permet conèixer la integritat del text xifrat, ja que qualsevol modificació d'aquest text, farà que el resum obtingut sigui diferent.

4.2 Interfície gràfica

Aquesta aplicació seguint els objectius del TFC establerts inicialment disposa d'una interfície gràfica d'usuari. Des d'un començament s'ha dissenyat amb la idea de fer una interfície fàcil d'utilitzar i molt amigable, i que a la vegada mantingui informat a l'usuari dels processos que es porten a terme en tot moment.

Després de la seva implementació es pot dir que aquesta interfície gràfica té les següents característiques:

a) Sensació de control: els usuaris són els que decideixen en tot moment què s'ha de fer.

b) Familiaritat per a l'usuari: l'aplicació fa servir poca terminologia especialitzada.

c) Temps de resposta adequat: tot i que el temps de resposta de l'aplicació depèn de la longitud de l'arxiu a xifrar, en les proves realitzades no ha estat massa llarg, encara que de totes maneres, l'aplicació sempre manté informat a l'usuari de l'operació que s'està realitzant.

d) Atenció a la possibilitat d'errors d'usuari.

e) Feedback immediat a l'usuari: l'usuari se n'adona del fet que la seva acció ha estat acceptada o rebutjada o bé si ja s'ha completat o encara no.

f) Robustesa: amb les proves efectuades, quasi es pot afirmar que, faci el que faci l'usuari, l'aplicació no es descontrolarà.

g) Facilitat d'utilització.

h) Facilitat d'aprenentatge.

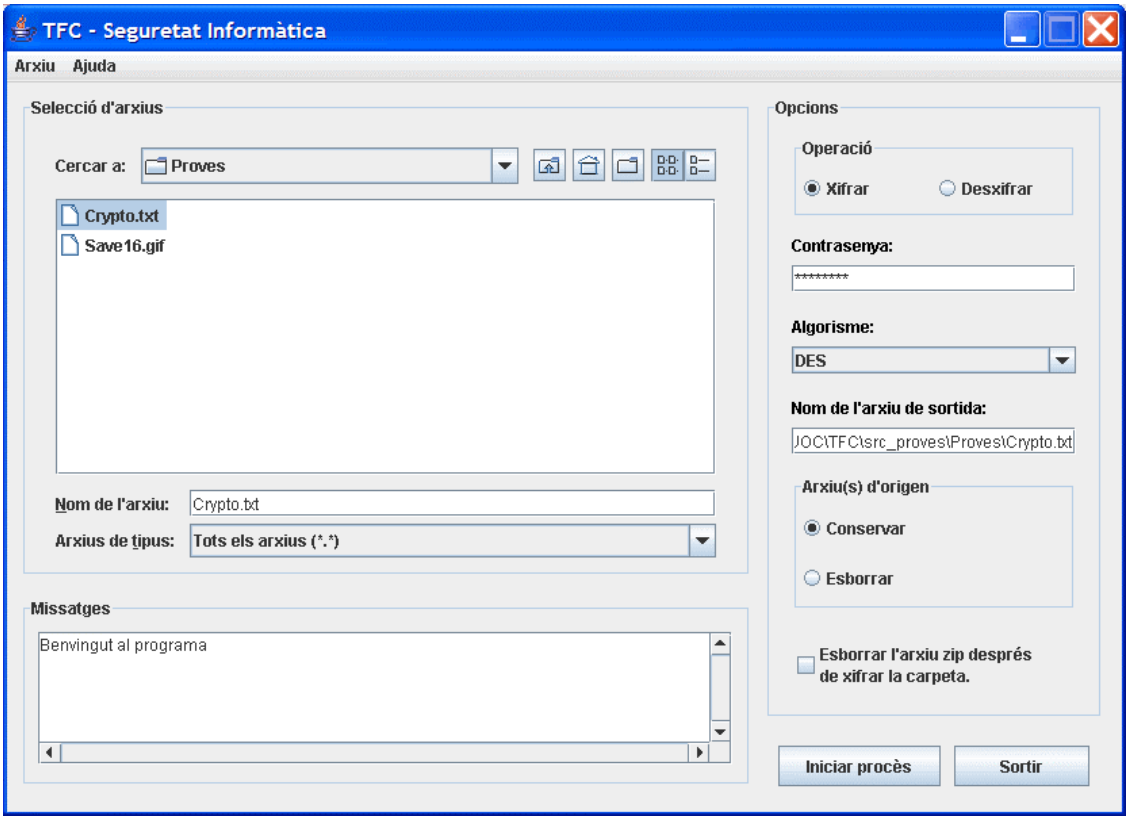


Figura 9. Visió general de la interfície gràfica

Desenvolupar la interfície gràfica d'usuari ha estat relativament senzill tot i que laboriós gràcies a la utilització dels components proporcionats per Java en el seu paquet Swing.

Es pot veure que s'utilitzen una gran varietat d'aquests components, finestres, diàlegs, menús, selectors d'arxius (file chooser's), botons, radio botons, quadres de selecció, quadres de text, llistes desplegables.

El fet que durant la carrera no s'hagi vist amb profunditat el disseny i la implementació d'interfícies gràfiques d'usuari limita molt el coneixement i la utilització d'aquestes tècniques i ha requerit d'una major dedicació de temps.

Especialment difícil ha estat la integració del selector d'arxius i el seu correcte funcionament. Per exemple el fet d'aconseguir que la interfície estigués en Català va requerir d'un procés de recerca de moltes hores. També cal dir que aquest component té certs *bugs* de funcionament que fan que per exemple un usuari hagi de prémer la tecla intro després d'escriure el nom d'un fitxer per tal

de confirmar la seva selecció, és a dir, que internament l'objecte tingui aquell arxiu com l'arxiu escollit.

4.3 Treball amb carpetes

Per poder treballar amb carpetes en un entorn gràfic, el primer pas ha estat la recerca d'un objecte gràfic de l'API de Java que permeti realitzar aquest procés de la forma més simple possible. Aquest objecte és el JFileChooser, amb l'opció fc.setFileSelectionMode(JFileChooser.FILES_AND_DIRECTORIES); que permet seleccionar arxius o directoris independentment.

Si es selecciona un arxiu, llavors només caldrà xifrar aquest arxiu. En canvi si hem seleccionat un directori, el procés a realitzar serà comprimir tots els arxius que hi ha en la carpeta en un arxiu del tipus Zip i posteriorment es xifrarà aquest arxiu com en el cas anterior.

Per poder comprimir la carpeta, en primer lloc cal obtenir el nom de tots els arxius que es troben a la carpeta i a les seves subcarpetes. Aquest procés es fa de forma recursiva amb la funció obtenirNomsArxius de la classe Utilitats. Una vegada es té una array amb el nom de tots els arxius (sense carpetes) es passa a comprimir-los i es desen en un arxiu que té com a nom el nom de la carpeta original amb l'extensió zip i que es desarà a la carpeta arrel on es trobava la carpeta que hem acabat de comprimir.

L'usuari pot escollir si una vegada xifrat l'arxiu zip aquest es vol esborrar, de la mateixa manera com es pot esborrar la carpeta amb els fitxers originals.

En el procés de desxifrat, únicament es desxifra l'arxiu obtenint com a resultat l'arxiu zip que conté tots els arxius originals. Posteriorment l'usuari haurà de descomprimir aquest arxiu utilitzant alguna de les aplicacions comercials de descompressió.

Comprimir la carpeta abans de xifrar els arxius era una de les possibilitats. També es va valorar la possibilitat de xifrar cada arxiu per separat.

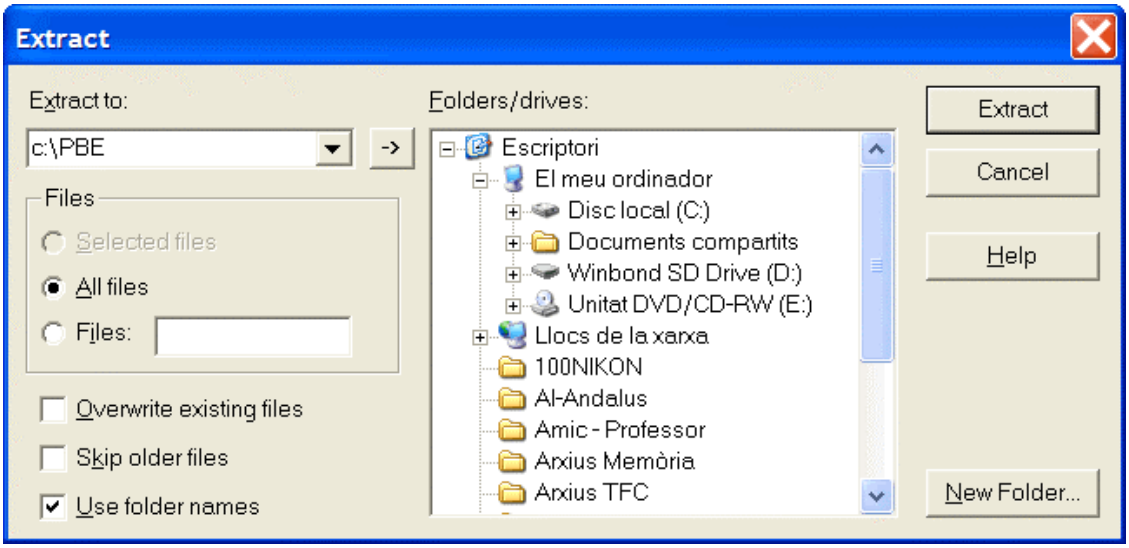
Al final es va optar per la primera opció ja que d'aquesta manera es millora la velocitat de l'aplicació i es minimitza l'espai de disc utilitzat.

5 Manual d'instal·lació

Aquesta aplicació per poder funcionar correctament necessita tenir instal·lada la versió 5 de la plataforma Java, disponible a l'espai de descàrregues Java de Sun Microsystems, http://java.sun.com/j2se/1.5.0/download.jsp.

No es pot assegurar el correcte funcionament de l'aplicació en versions anteriors de Java, ja que un dels algorismes de xifratge utilitzats no està suportat per aquestes versions.

La resta dels arxius necessaris per al funcionament de l'aplicació estan inclosos en l'arxiu jcabellos_PBE.zip. Aquest arxiu està comprimit en format Zip, i per tant el primer pas a realitzar serà descomprimir-lo en una carpeta qualsevol. S'aconsella conservar l'estructura de carpetes dels fitxers inclosos en aquest arxiu comprimit (Use Folder Names).



Es crearà un arbre de carpetes amb l'arrel anomenada PBE, el contingut del qual s'especifica més endavant.

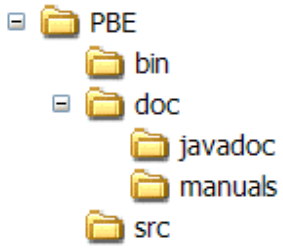


Figura 10. Arbre de carpetes

- carpeta bin: conté els arxius executables (.class) de l'aplicació.

- carpeta doc\manuals: conté els manuals d'instal·lació i d'usuari.

- carpeta doc\javadoc: conté els arxius de documentació generats amb javadoc necessaris per a la possible reutilització de les classes per part de qualsevol desenvolupador en altres versions i/o projectes.

- carpeta src: conté els arxius font de l'aplicació.

- Un arxiu anomenat PBE.bat que permet arrencar el programa directament des de l'explorador.

Una vegada acabat el procés de descompressió, el programa es podrà executar de diverses maneres:

- Executant el fitxer PBE.bat des del explorador del Windows. Aquest arxiu el que fa és executar l'ordre java PBE que iniciarà l'execució de la interfície gràfica del programa.

- Executant l'ordre java PBE_GUI des de dins d'una finestra de l'indicador d'ordres del DOS situada a la carpeta PBE\bin.

6 Manual d'usuari

6.1 Introducció

PBE és una aplicació Windows de 32 bits que permet el xifratge i el desxifratge d'arxius i carpetes a partir d'una contrasenya introduïda per l'usuari.

La versió actual de l'aplicació permet xifrar utilitzant les especificacions PBEWithMD5AndDES i PBEWithSHA1AndDESede pels algorismes DES i Triple Des respectivament.

Aquesta versió dóna suport al xifratge de carpetes, comprimint el contingut en un arxiu de tipus zip. L'arxiu comprimit es pot conservar o eliminar a elecció de l'usuari després de xifrar el contingut de la carpeta.

Està dissenyat amb una interfície gràfica molt amigable, intuïtiva i fàcil d'aprendre a fer funcionar.

6.2 La finestra principal

La interfície gràfica d'usuari presenta una finestra principal amb tres zones ben diferenciades.

La primera zona correspon al selector d'arxius o carpetes. És el típic arbre d'arxius amb els que estem acostumats a treballar en el sistema operatiu Windows. Es pot seleccionar qualsevol arxiu o carpeta que formi part de l'equip. En cas de seleccionar una carpeta, aquesta es comprimirà abans de començar el procés de xifrat.

La segona zona correspon al conjunt d'opcions que l'usuari pot escollir:

2 botons per activar la forma de treballar
- Xifrar – per xifrar arxius
- Desxifrar – per desxifrar arxius

1 quadre de text per introduir la contrasenya
1 caixa desplegable per escollir l'algorisme a aplicar
1 quadre de text per introduir la contrasenya
2 botons per establir el criteri a seguir amb els arxius originals
- Conservar – per mantenir l'arxiu original
- Esborrar – per esborrar l'arxiu original

1 botó per establir el criteri a seguir amb l'arxiu comprimit en cas d'haver seleccionat xifrar una carpeta.

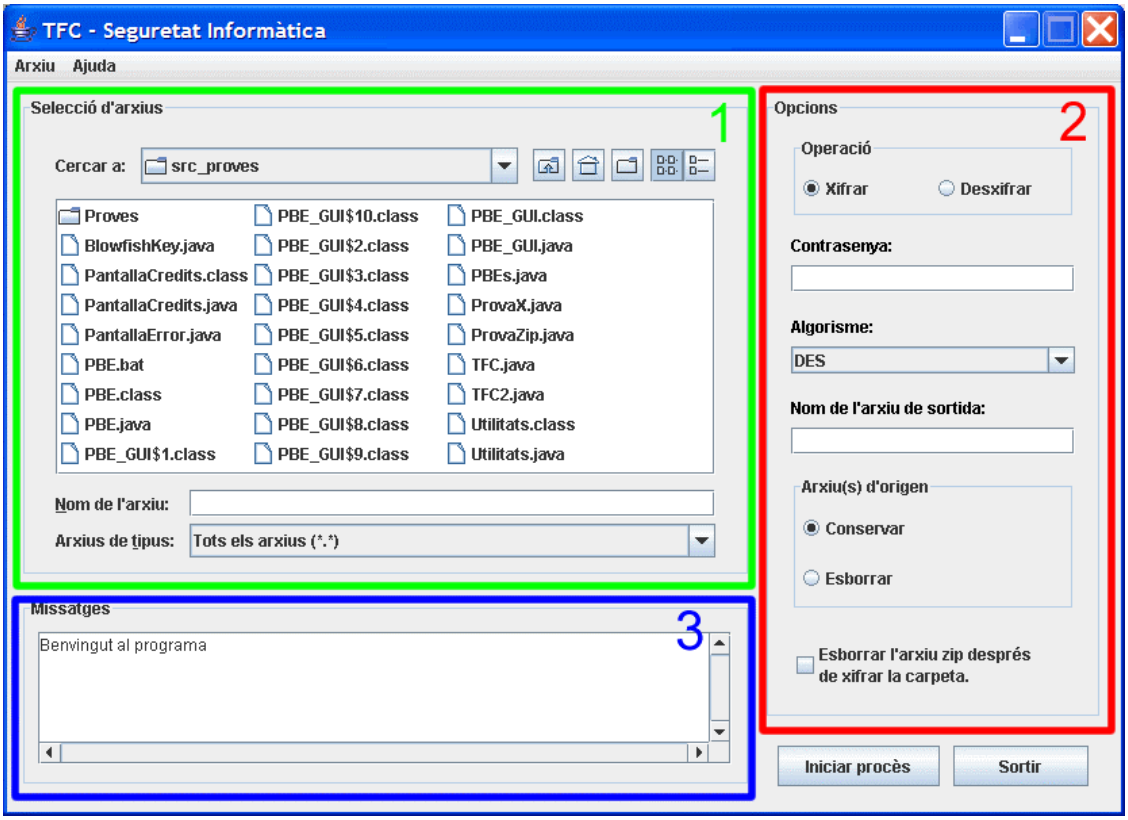


Figura 11. Finestra principal de l'aplicació

La tercera zona correspon a un àrea d'informació per l'usuari, de forma que conegui en tot moment en quin estat del procés es troba en aquell moment, o en cas de produir-se un error quina ha estat la causa del mateix. Aquesta informació es manté en la finestra i es pot revisar mentre dura la sessió de treball actual.

6.3 Xifrar un arxiu

Per xifrar un arxiu cal seguir el següent procediment:

1. Escollir l'arxiu o la carpeta a xifrar.
2. Seleccionar el botó Xifrar.
3. Escriure la contrasenya en el camp destinat a tal fi.
4. Escollir l'algorisme a utilitzar en la caixa desplegable.
5. Introduir el nom de l'arxiu de sortida. Per defecte, l'arxiu tindrà el mateix nom que l'arxiu d'origen.

6. Escollir que es vol fer amb els arxius d'origen (conservar/eliminar).

7. Si es tracta d'una carpeta seleccionar el botó per eliminar l'arxiu comprimit o bé no seleccionar-lo per conservar-lo.

8. Finalment prémer el botó **Iniciar procés** per començar el procediment de xifrat.

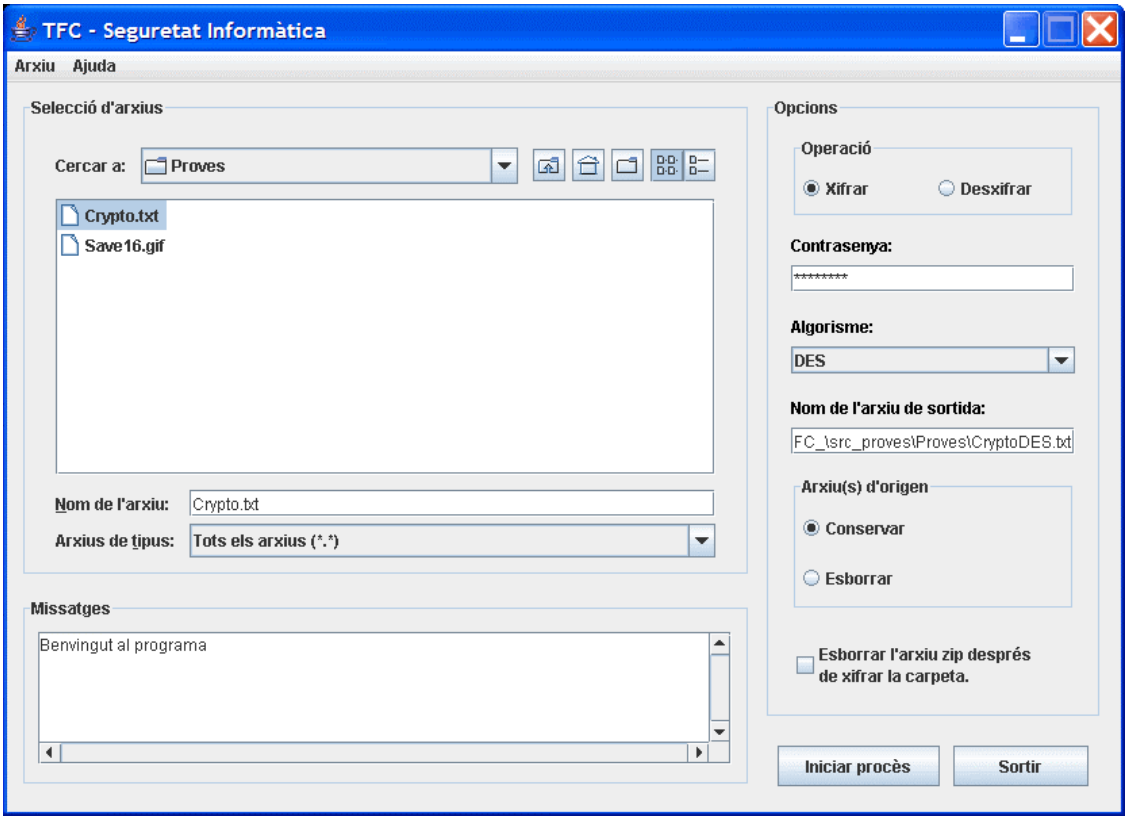


Figura 12. Procés de xifratge.

Si el procés finalitza amb èxit, rebrem una confirmació per pantalla.

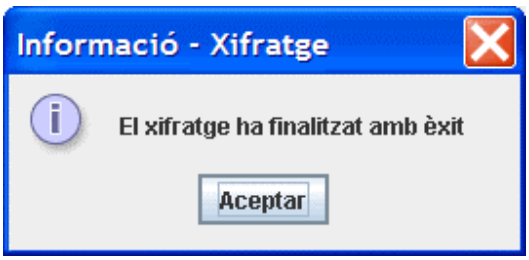


Figura 13. Missatge d'èxit en el xifratge

6.4 Desxifrar un arxiu

Per desxifrar un arxiu cal segui el següent procediment:

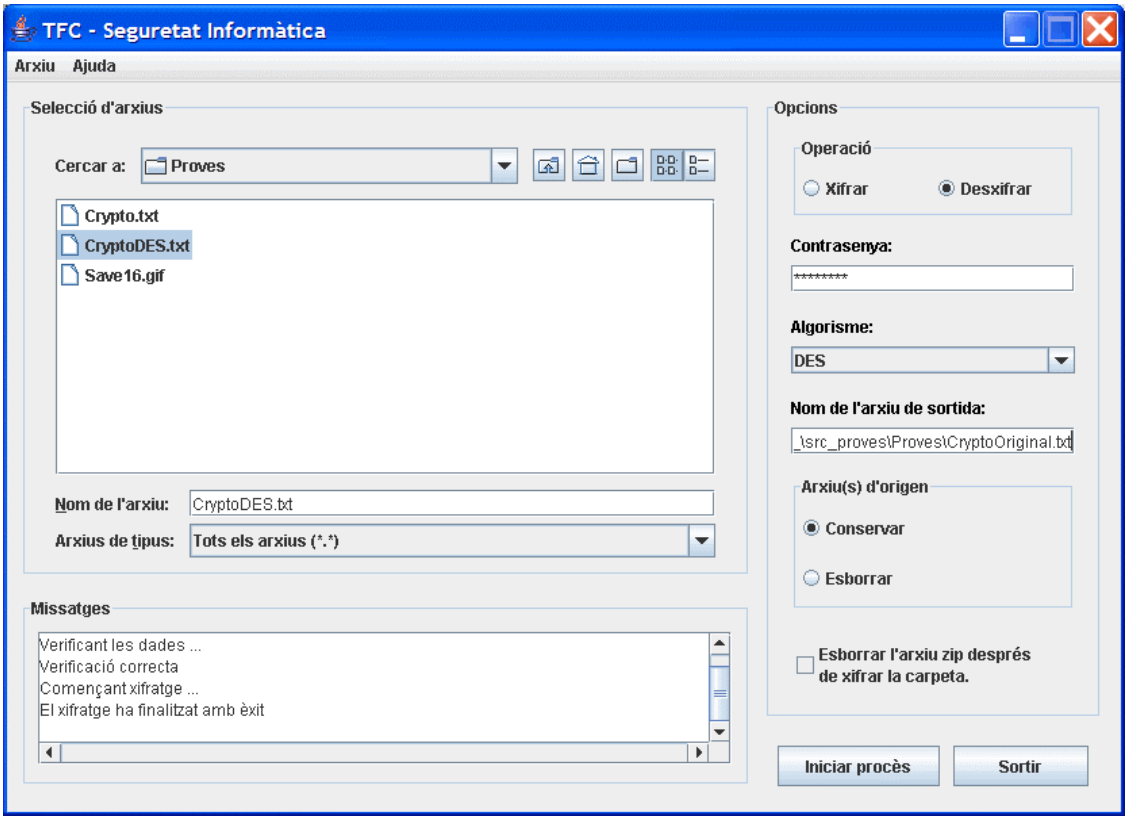


Figura 14. Procés de desxifratge

1. Escollir l'arxiu o la carpeta a desxifrar.
2. Seleccionar el botó Desxifrar.
3. Escriure la contrasenya en el camp destinat a tal fi.
4. Escollir l'algorisme a utilitzar en la caixa desplegable.
9. Introduir el nom de l'arxiu de sortida. Per defecte, l'arxiu tindrà el mateix nom que l'arxiu d'origen.
5. Escollir que es vol fer amb els arxius d'origen (conservar/eliminar).
6. Finalment prémer el botó **Iniciar procés** per començar el procediment de desxifrat.

Si el procés finalitza amb èxit, rebrem una confirmació per pantalla.

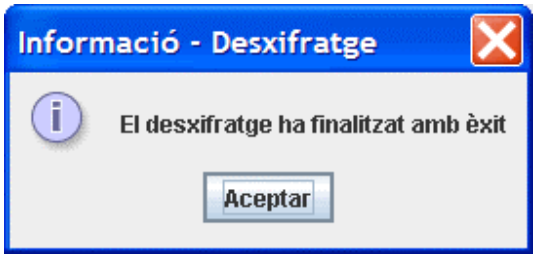


Figura 15. Missatge d'èxit en el desxifratge

6.5 Finalitzar el programa

El programa permet la seva finalització de tres formes diferents:

- Prement el botó Sortir.
- Seleccionant l'opció Sortir del menú Arxiu.
- Tancant la finestra amb el botó  .

6.6 Errades freqüents

Les errades més freqüents són degudes a que o bé no s'han informat tots els camps obligatoris en els processos de xifratge i/o desxifratge o bé alguns d'ells tenen un valor incorrecte.

Alguns exemples d'errors comuns són:

- longitud incorrecta de la contrasenya (mínim 8 caràcters)
- introducció de caràcters no vàlids (no ASCII) en la contrasenya, com per exemple el caràcter 'ñ'
- no seleccionar l'arxiu a xifrar o desxifrar
- no indicar cap nom per l'arxiu de sortida
- seleccionar una carpeta com a arxiu de sortida
- seleccionar xifrar una carpeta buida

En tots aquests casos, es rebrà un missatge per pantalla amb indicació de l'error comés.

La següent imatge mostra l'error que s'ha produït en la verificació de les dades per un procés de xifratge en el qual no s'han informat tots els camps.

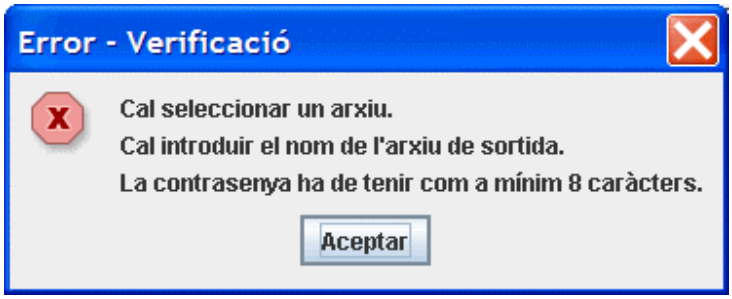


Figura 16. Exemple de missatge d'error

7 Proves realitzades

Per comprovar la validesa de l'aplicació s'han realitzat un conjunt de proves que han demostrat la seva validesa, tot i que no es poden descartar problemes de funcionament no previstos en les proves realitzades.

El conjunt de proves realitzades es poden classificar en:

- Proves relacionades amb l'interfície gràfica. Comprovar que tots els botons de la interfície responen a la seva funcionalitat, incloent-hi com els botons d'opció i/o de selecció. De la mateixa manera comprovar que totes les informacions dirigides cap a l'usuari es llegeixen de forma correcta.

- Forçar els errors que s'hagi previst en els blocs try-catch per comprovar el funcionament correcte del programari.

- Introduir contrasenyes de diferents longituds i comprovar la seva validesa tant a nivell de longitud (nombre de caràcters) com a la seva lògica (que no donin lloc a cap error de invalidesa de la clau).

- Seleccionar diferents arxius i/o directoris i xifrar-los.

- Seleccionar diferents arxius xifrats i desxifrar-los. Proves realitzades amb arxius xifrats correctament i amb una contrasenya correcta, amb arxius xifrats correctament i amb una contrasenya incorrecta, amb arxius xifrats modificats posteriorment i amb una contrasenya correcta (comprovar la validesa a partir de la signatura digital implementada).

El conjunt de proves realitzades es documenta amb les següents fitxes en les que es pretén verificar que l'aplicació realitza correctament totes les accions esperades o si més no si no s'aconsegueix en una primera fase, tenir documentats els errors per a posteriors revisions.

Subsistema:	**Interacció amb l'usuari**		
Realitzat per:	José Cabello Sánchez		
Propòsit:	Interacció de l'usuari amb l'aplicació		
Nº	**Entrada**	**Acció esperada**	**Verif.**
1	No es selecciona cap arxiu i es prem el botó per iniciar el xifratge/desxifratge	Error: Obtenim el missatge "Cal seleccionar un arxiu".	V
2	No s'introdueix cap contrasenya.	Error: Obtenim el missatge "La contrasenya ha de tenir com a mínim 8 caràcters".	V
3	La contrasenya inclou caràcters no vàlids (ñ)	Error: Obtenim el missatge "La contrasenya conté caràcters no vàlids (ñ)".	V

4	No s'informa el nom de l'arxiu de sortida	Error: Obtenim el missatge "Cal introduir el nom de l'arxiu de sortida".	**V**
5	El nom de l'arxiu de sortida és un directori	Error: Obtenim el missatge "El nom de l'arxiu de sortida és un directori".	**V**
6	Es selecciona un directori sense fitxers	Error: Obtenim el missatge "La carpeta seleccionada és buida"	**V**
7	Es selecciona un arxiu inexistent	Error: Obtenim el missatge "L'arxiu seleccionat no existeix"	**V**
8	S'escriu el nom d'un arxiu d'entrada i no es confirma la seva selecció prement Intro	Error: Obtenim el missatge "El nom de l'arxiu informat no ha estat confirmat, cal prémer Intro al quadre de text"	**V**
9	El nom de l'arxiu d'entrada i el nom de l'arxiu de sortida coincideixen.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? Possibles respostes SI NO	**V**
10	Combinacions dels errors anteriors.	Error: Obtenim un missatge combinació de tots els errors comesos	**V**
11	L'usuari tanca l'aplicació a través del botó Sortir	Es tanca l'aplicació	**V**
12	L'usuari tanca l'aplicació a través del menú Arxiu/Sortir	Es tanca l'aplicació	**V**
13	L'usuari tanca l'aplicació a través del botó X	Es tanca l'aplicació	**V**

Subsistema:	**Xifratge d'arxius**		
Realitzat per:	José Cabello Sánchez		
Propòsit:	L'aplicació xifra un arxiu que conté un arxiu amb el text en clar utilitzant un dels algorismes implementats. (Verificació de les opcions)		
Nº	**Entrada**	**Acció esperada**	**Verif.**
1	Xifratge d'un arxiu amb l'algorisme DES, conservant els arxius d'origen i amb un nom de l'arxiu de sortida diferent del d'entrada i no existent.	Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit"	**V**
2	Xifratge d'un arxiu amb l'algorisme Triple DES, conservant l'arxiu d'origen i amb un nom de l'arxiu de sortida diferent del d'entrada i no existent.	Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit"	**V**
3	Xifratge d'un arxiu amb l'algorisme DES, conservant l'arxiu d'origen i amb un nom de l'arxiu de sortida diferent del d'entrada i que ja existeix.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit" NO -> Finalitza l'execució del programa	**V**
4	Xifratge d'un arxiu amb l'algorisme Triple DES, conservant l'arxiu d'origen i amb un nom de l'arxiu de sortida diferent del d'entrada i que ja existeix.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit" NO -> Finalitza l'execució del programa	**V**

5	Xifratge d'un arxiu amb l'algorisme DES, conservant l'arxiu d'origen i amb el mateix nom que l'arxiu de sortida.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit" NO -> Finalitza l'execució del programa	**V**
6	Xifratge d'un arxiu amb l'algorisme Triple DES, conservant l'arxiu d'origen i amb el mateix nom que l'arxiu de sortida.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit" NO -> Finalitza l'execució del programa	**V**
7	Xifratge d'un arxiu amb l'algorisme DES, eliminant l'arxiu d'origen i amb un nom per l'arxiu de sortida existent.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit" i els arxius d'origen s'han esborrat NO -> Finalitza l'execució del programa	**V**
8	Xifratge d'un arxiu amb l'algorisme Triple DES, eliminant l'arxiu d'origen i amb un nom per l'arxiu de sortida existent.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit" i els arxius d'origen s'han esborrat NO -> Finalitza l'execució del programa	**V**
7	Xifratge d'un arxiu amb l'algorisme DES, eliminant l'arxiu d'origen i amb un nom per l'arxiu de sortida no existent.	Els arxius d'origen s'han esborrat Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit"	**V**
8	Xifratge d'un arxiu amb l'algorisme Triple DES, eliminant l'arxiu d'origen i amb un nom per l'arxiu de sortida no existent.	Els arxius d'origen s'han esborrat Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit"	**V**

Subsistema:	**Desxifratge**		
Realitzat per:	José Cabello Sánchez		
Propòsit:	L'aplicació desxifra un arxiu que conté un arxiu xifrat prèviament utilitzant un dels algorismes implementats. (Verificació de les opcions)		
Nº	**Entrada**	**Acció esperada**	**Verif.**
1	Desxifratge d'un arxiu amb l'algorisme DES, conservant l'arxiu d'origen i amb un nom de l'arxiu de sortida diferent del d'entrada i no existent. La contrasenya és correcta.	Informació: Obtenim el missatge "El desxifratge ha finalitzat amb èxit"	**V**

2	Desxifratge d'un arxiu amb l'algorisme Triple DES, conservant l'arxiu d'origen i amb un nom de l'arxiu de sortida diferent del d'entrada i no existent. La contrasenya és correcta.	Informació: Obtenim el missatge "El desxifratge ha finalitzat amb èxit"	**V**
3	Desxifratge d'un arxiu amb l'algorisme DES, conservant l'arxiu d'origen i amb un nom de l'arxiu de sortida diferent del d'entrada i que ja existeix. La contrasenya és correcta.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El desxifratge ha finalitzat amb èxit" NO -> Finalitza l'execució del programa	**V**
4	Desxifratge d'un arxiu amb l'algorisme Triple DES, conservant l'arxiu d'origen i amb un nom de l'arxiu de sortida diferent del d'entrada i que ja existeix. La contrasenya és correcta.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El desxifratge ha finalitzat amb èxit" NO -> Finalitza l'execució del programa	**V**
5	Desxifratge d'un arxiu amb l'algorisme DES, conservant l'arxiu d'origen i que té el mateix nom que l'arxiu de sortida. La contrasenya és correcta.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El desxifratge ha finalitzat amb èxit" NO -> Finalitza l'execució del programa	**V**
6	Desxifratge d'un arxiu amb l'algorisme Triple DES, conservant l'arxiu d'origen i que té el mateix nom que l'arxiu de sortida. La contrasenya és correcta.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El desxifratge ha finalitzat amb èxit" NO -> Finalitza l'execució del programa	**V**
7	Desxifratge d'un arxiu amb l'algorisme DES, eliminant l'arxiu d'origen i amb un nom per l'arxiu de sortida existent. La contrasenya és correcta.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El desxifratge ha finalitzat amb èxit" i els arxius d'origen s'han esborrat NO -> Finalitza l'execució del programa	**V**
9	Desxifratge d'un arxiu amb l'algorisme Triple DES, eliminant l'arxiu d'origen i amb un nom per l'arxiu de sortida existent. La contrasenya és correcta.	Informació: Obtenim el missatge "Ja existeix un arxiu amb el nom ARXIU.NOM. Desitja sobreescriure'l? SI -> Informació: Obtenim el missatge "El desxifratge ha finalitzat amb èxit" i els arxius d'origen s'han esborrat NO -> Finalitza l'execució del programa	**V**
10	Desxifratge d'un arxiu amb l'algorisme DES, eliminant l'arxiu d'origen i amb un nom per l'arxiu de sortida no existent. La contrasenya és correcta.	Els arxius d'origen s'han esborrat Informació: Obtenim el missatge "El desxifratge ha finalitzat amb èxit"	**V**

11	Desxifratge d'un arxiu amb l'algorisme Triple DES, eliminant els arxius d'origen i amb un nom per l'arxiu de sortida no existent. La contrasenya és correcta.	Els arxius d'origen s'han esborrat Informació: Obtenim el missatge "El desxifratge ha finalitzat amb èxit"	**V**
12	Desxifratge d'un arxiu xifrat amb l'algorisme DES, utilitzant l'algorisme correcte i amb una contrasenya incorrecta.	Error: Obtenim el missatge "S'ha produït un error durant el xifratge/desxifratge. Probablement no s'utilitza l'algorisme o contrasenya correcte" + Informació de la excepció.	**V**
13	Desxifratge d'un arxiu xifrat amb l'algorisme Triple DES, utilitzant l'algorisme correcte i amb una contrasenya incorrecta.	Error: Obtenim el missatge "S'ha produït un error durant el xifratge/desxifratge. Probablement no s'utilitza l'algorisme o contrasenya correcte" + Informació de la excepció.	**V**
14	Desxifratge d'un arxiu xifrat amb l'algorisme DES, utilitzant l'algorisme incorrecte i amb una contrasenya correcta.	Error: Obtenim el missatge "S'ha produït un error durant el xifratge/desxifratge. Probablement no s'utilitza l'algorisme o contrasenya correcte" + Informació de la excepció.	**V**
15	Desxifratge d'un arxiu xifrat amb l'algorisme Triple DES, utilitzant l'algorisme incorrecte i amb una contrasenya correcta.	Error: Obtenim el missatge "S'ha produït un error durant el xifratge/desxifratge. Probablement no s'utilitza l'algorisme o contrasenya correcte" + Informació de la excepció.	**V**
16	Desxifratge d'un arxiu xifrat posteriorment modificat utilitzant l'algorisme i la contrasenya correctes. L'arxiu té la mateixa longitud que l'original i no es modifica el *padding*.	Error: Obtenim el missatge "L'arxiu ha estat modificat".	**V**
17	Desxifratge d'un arxiu xifrat posteriorment modificat utilitzant l'algorisme i la contrasenya correctes. L'arxiu té una longitud que no és múltiple de 8.	Error: Obtenim el missatge "S'ha produït un error durant el xifratge/desxifratge. Probablement l'arxiu ha estat modificat". + Informació de la excepció.	**V**
18	Desxifratge d'un arxiu xifrat posteriorment modificat utilitzant l'algorisme i la contrasenya correctes. L'arxiu té una longitud que és múltiple de 8 i s'ha modificat el *padding*.	Error: Obtenim el missatge "S'ha produït un error durant el xifratge/desxifratge. Probablement l'arxiu ha estat modificat". + Informació de la excepció.	**V**
19	Desxifratge d'un arxiu xifrat posteriorment modificat utilitzant l'algorisme i la contrasenya correctes. L'arxiu té una longitud diferent de la inicial, que és múltiple de 8 i no s'ha modificat el *padding*.	Error: Obtenim el missatge "S'ha produït un error durant el xifratge/desxifratge. Probablement l'arxiu ha estat modificat". + Informació de la excepció.	**V**

Subsistema:	**Xifratge de carpetes**
Realitzat per:	José Cabello Sánchez
Propòsit:	L'aplicació crea un arxiu zip amb el contingut de la carpeta. Posteriorment aquest arxiu amb el text en clar és xifrat utilitzant un dels algorismes implementats. (Veure subsistema xifratge d'arxius)

Nº	**Entrada**	**Acció esperada**	**Verif.**
1	Xifratge d'una carpeta amb els algorismes DES o Triple DES, conservant els arxius d'origen i el fitxer zip.	Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit"	**V**
2	Xifratge d'una carpeta amb l'algorismes DES o Triple DES, conservant els arxius d'origen i esborrant el fitxer zip.	Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit" L'arxiu zip s'ha esborrat.	**V**
3	Xifratge d'una carpeta amb l'algorismes DES o Triple DES, esborrant els arxius d'origen i conservant el fitxer zip.	Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit" Els arxius d'origen s'han esborrat	**V**
4	Xifratge d'una carpeta amb l'algorismes DES o Triple DES, esborrant els arxius d'origen i esborrant el fitxer zip.	Informació: Obtenim el missatge "El xifratge ha finalitzat amb èxit" Els arxius d'origen i l'arxiu zip s'han esborrat.	**V**

8 Comentaris i conclusions

Avui dia, el disseny dels aspectes relacionats amb la seguretat informàtica és un tema fonamental per a les aplicacions informàtiques i s'ha d'entendre com una part més de l'aplicació, i no com un afegitó que es desenvoluparà com a complement al programari original.

En aquest context, és convenient seguir els estàndards d'implementació dels algorismes criptogràfics, ja que en facilita la programació i minimitza el temps de manteniment de les aplicacions.

Des d'aquest punt de vista, la conclusió principal a la que podem arribar és que hem estat capaços de desenvolupar una aplicació que compleix els objectius plantejats inicialment, ja que permet el xifratge d'arxius de forma segura, utilitzant una interfície gràfica per a la interacció amb l'usuari. A més es una aplicació robusta, senzilla d'utilitzar, amb un temps de resposta adequat i que presenta un alt nivell de feedback amb l'usuari. Així mateix, el fet d'utilitzar la llibreria criptogràfica de Java permet simplificar la instal·lació de l'aplicació, ja que no és necessari la difícil instal·lació de les llibreries criptogràfiques d'altres proveïdors.

D'altra banda, també s'ha pogut experimentar la dificultat de realitzar una correcta planificació temporal en els projectes informàtics sense que apareguin desviacions, com demostra el fet que el cost de desenvolupament (temps) de l'aplicació ha estat superior al previst inicialment.

També és interessant comentar alguns dels beneficis acadèmics aportats pel treball final de carrera, com:

- Superació del repte que suposa treballar en un tema que es troba en la base de moltes aplicacions actuals.
- Aplicació de múltiples procediments i conceptes treballats en diverses assignatures fent possible un millor assoliment dels mateixos.
- Introducció a nous elements del llenguatge Java (Swing).

Per últim, és important indicar algunes idees per a possibles línies de millora i ampliacions, com:

- Implementació d'un mòdul complementari de signatura digital.
- Eliminació permanent dels arxius esborrats sobreescrivint els sectors del disc afectats.
- Selecció múltiple d'arxius.

9 Glossari

AES: Advanced Encryption Standard. Criptosistema Rijndael, que xifra blocs de 128 bits per mitjà d'una clau que pot variar la longitud entre 128, 192 o 256 bits.

Atac: estratègia o mètode que té per objectiu descobrir la clau de xifratge o bé el text en clar. Els atacs criptoanalítics exploten les febleses dels algorismes de xifra. Els atacs als sistemes informàtics i de comunicacions exploten les vulnerabilitats d'aquests sistemes.

Autenticitat: propietat de trobar-se, en relació amb la informació, en el mateix estat en què va ser produïda, sense modificacions no autoritzades; és sinònim d'integritat.

Autentificació: comprovació de l'autenticitat.

CBC: Cipher Bloc Chaining. Mode de xifratge de bloc en què 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.

CFB: Cipher Feedback. Mode de xifratge de bloc en què la llargada dels blocs de text no ha de coincidir amb la dels blocs del criptosistema.

Clau: paràmetre, normalment secret, que controla els processos de xifratge i/o de desxifratge.

Complexitat lineal d'una seqüència: nombre de cel·les de l'LFSR més curt que és capaç de generar una seqüència.

Criptoanàlisi: ciència que s'ocupa de trencar xifres, és a dir, descobrir la clau o el text en clar usats com a entrades de la xifra.

Criptografia: ciència i estudi de l'escriptura secreta.

Criptograma: text xifrat.

Criptologia: denominació conjunta de la criptografia i de la criptoanàlisi.

Criptosistema: xifra.

Criptosistema de clau compartida: criptosistema en què tant l'emissor com el receptor
comparteixen una sola clau que fan servir tant per a xifrar com per a desxifrar.

Criptosistema de clau pública: criptosistema en què cada usuari té una clau pública i
una de privada; amb aquest criptosistema una parella d'usuaris es pot comunicar confidencialment sense compartir una clau secreta.

Criptosistema de flux: sistema de xifratge que utilitza un generador pseudoaleatori per a xifrar un missatge, sumant bit a bit el text en clar amb la seqüència pseudoaleatòria que resulta del generador.

DES: Data Encryption Standard. Criptosistema de xifratge de bloc que xifra blocs de dades de 64 bits de llargada per mitjà d'una clau de 56 bits i l'acció de caixes S.

Desxifratge: procés de transformació del text xifrat en text en clar.

ECB: Electronic Code Book. Mode de xifratge de bloc en què el xifratge dels blocs és independent l'un de l'altre i es porta a terme amb una mateixa clau.

Estat d'un LFSR: conjunt de valors continguts en cada cel·la d'un LFSR en un instant de temps.

Generador lineal: generador de seqüències de bits que només executa operacions lineals sobre els elements d'entrada per a obtenir la seqüència de sortida.

Generador no lineal: generador de seqüències de bits que executa operacions no lineals, com ara permutacions, sobre els elements d'entrada per a obtenir la seqüència de sortida; a més, pot fer servir també operacions lineals.

Generador pseudoaleatori: procés determinista capaç de generar una seqüència pseudoaleatòria.

Integritat: propietat de no haver sofert, en relació amb la informació, modificacions ni
supressions parcials no autoritzades.

LFSR: registre de desplaçament realimentat linealment.

NLFSR: registre de desplaçament realimentat no linealment.

OFB: Ouput Feedback. Mode de xifratge de bloc en què el vector inicial es realimenta directament amb el resultat del xifratge de bloc.

Polinomi de connexions d'un LFSR: polinomi que determina o que és determinat per
la funció lineal de realimentació de l'LFSR.

Privacitat: dret de les persones a salvaguardar la seva intimitat, especialment pel que fa a les dades de què disposen les entitats públiques o privades.

Registre de desplaçament realimentat linealment: dispositiu físic o lògic format per
n cel·les de memòria i una funció de realimentació lineal.

Seqüència de xifratge: seqüència que resulta del generador pseudoaleatori en un criptosistema de flux.

Seqüència pseudoaleatòria: seqüència que compleix els tres postulats de Golomb.

Triple DES: protocol de xifratge triple que utilitza el DES com a base per a obtenir un xifrador amb un espai de claus superior.

Xifra: mètode secret d'escriptura, mitjançant el qual un text en clar es transforma en un
text xifrat.

Xifra de substitució: xifra basada a canviar els bits o els caràcters del text en clar per substituts.

Xifra de transposició: xifra basada a reordenar els bits o els caràcters del text en clar.

Xifratge: procés de transformació d'un text en clar en un text xifrat.

10 Bibliografia i recursos utilitzats

10.1 Eines

- Llenguatge de programació: Java (JDK 1.5.0_06)
- Sistema Operatiu de desenvolupament: Windows XP Profesional
- IDE de desenvolupament: JCreator
- Llibreries criptogràfiques: Les proporcionades per Sun Microsystems a l'API de Java.
- Editor de textos: Microsoft Word + PDF Creator
- Editor de presentacions: Microsoft PowerPoint

10.2 Bibliografia

Criptografia. J. Domingo, J. Herrera, H. Rifà. Barcelona. UOC 2004.

Data Encryption Standard (DES), **FIPS PUB 46-3.** National Institute of Standards and Technology (NIST). Octubre 1999.

Handbook of Applied Cryptography, A. Menezes, P. van Oorschot, S. Vanstone, CRC Press, 1996. www.cacr.math.uwaterloo.ca/hac

Java Criptography. Jonathan B. Knudsen. OReilly 1998

Lecture Notes on Cryptography. S. Goldwasser, M. Bellare. Agost 2001 www.cse.ucsd.edu/users/mihir/papers/gb.html

PKCS#5: Password Based Encryption Standard. RSA Laboratories. Versió 2.0, Març 1999.

PKCS#12: Personal Information Exchange Syntax Standard. RSA Laboratories. Versió 1.0, Juny 1999.

Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, SP 800-67. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Maig 2004 Versió 1.

Técnicas criptográficas de protección de datos. A. Fúster, D. de la Guía, L. Hernández, F. Montoya, J. Muñoz. Madrid. Ra-Ma 1997.

Técniques de desenvolupament de programari. F. Xhafa. Barcelona. UOC 2002.