Seguretat informàtica

Memòria

Xifratge d'arxius

Alumne:	***Joan Benavent Mora***
Consultor:	*Antoni Martínez Ballesté*
Curs 2006/07, semestre de primavera	

RESUM

Un criptosistema o sistema criptogràfic consta esencialment d'aquests quatre elements:

→ El text que es vol xifrar, o text en clar (*plaintext*).
→ L'algorisme de xifrat, també anomenat xifra.
→ Una clau de xifratge (*key*).
→ El text xifrat (*ciphertext*).

La criptografia moderna ha heretat de la criptografia clàssica les dues tècniques més bàsiques per al xifratge d'un text: la **substitució** i la **transposició**. La primera consisteix en substituir per uns altres, les lletres, dígits o símbols del text en clar, conforme a un cert conjunt de regles prèviament fixades; la segona, la transposició, reordena dins el mateix text les lletres, dígits o símbols.

Els criptosistemes es classifiquen en dos grans grups:

a) **Xifres de clau simètrica** (o de clau compartida): per al desxifratge del text xifrat cal usar la mateixa clau amb la que ha estat xifrat el text en clar.
b) **Xifres de clau asimètrica** (o de clau pública): les claus emprades per a xifrar i desxifrar són diferents. La relació mútua que mantindran aquestes dues claus, però, s'expressarà en forma d'una certa equació matemàtica que s'haurà de complir.

Per al xifratge d'arxius, problema sobre el que he desenvolupat aquest projecte, la criptografia que s'utilitza es la de clau simètrica. Tractaré, per tant, només sobre qüestions relacionades amb aquesta tècnica de xifratge. La criptografia de clau simètrica es divideix, al seu torn, en dos subgrups: **xifres de bloc** i **xifres de flux**. El problema del xifratge d'arxius es resol, principalment, mitjançant algorismes de criptografia de bloc. En aquest cas, les xifres de flux complementen l'anterior, ja que, tal com he fet en implementar l'aplicació, poden usar-se per a la generació de les claus.

ÍNDEX DE CONTINGUTS

ÍNDEX DE FIGURES

1 Introducció

1.1 Justificació del TFC i context en el qual es desenvolupa

Actualment l'ús de la criptografia ha arribat a ser del tot generalitzat, tant en els processos de transmissió i intercanvi segur d'informació, com en l'emmagatzematge secret de dades. Es tracta d'una disciplina els fonaments teòrics de la qual són en l'Àlgebra i en el Càlcul de Probabilitats.

Havent cursat l'assignatura de *Criptografia*, la tria d'aquesta temàtica, *Seguretat informàtica,* per al TFC m'ha servit per consolidar part dels coneixements adquirits llavors, posant-los en pràctica mitjançant l'aplicació que he implementat. De les diferents propostes sobre les que hagués pogut basar el treball, he escollit el de *Xifratge d'arxius*.

El context en el qual he desenvolupat l'aplicació és el context propi d'un TFC dins la carrera d'Enginyeria Tècnica d'Informàtica de Sistemes. La relació mantinguda amb el professor que m'ha orientat i del qui he pogut disposar per a qualsevol pregunta o aclariment de dubtes no ha estat una relació personal directa igual com passa en les universitat presencials, sino a través del campus virtual de la UOC, la qual cosa, però, no ha representat cap inconvenient per a dur el projecte a bon terme.

A més de la materia especificament relacionada amb la seguretat, el TFC m'ha permés tenir un primer contacte amb la programació d'interfícies gràfiques en Java i amb la manipulació, tot i que molt elemental, de documents XML.

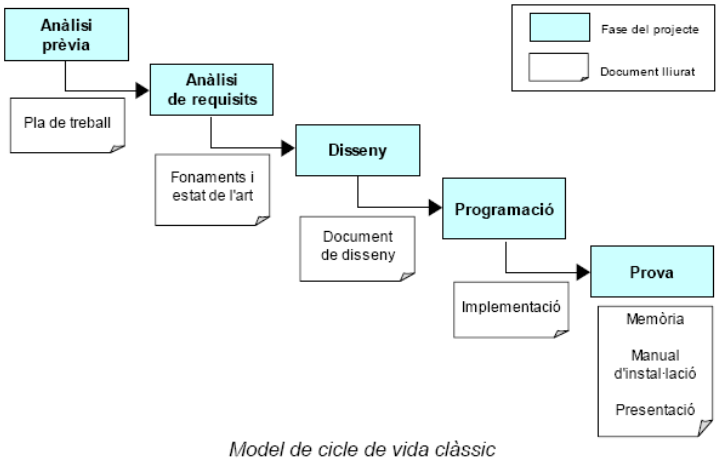
1.2 Objectius del TFC

El objectius que m'havia marcat amb el TFC han estat, entre d'altres, el següents:

- Plantejar un problema concret (xifratge d'arxius) i proposar-hi una solució.
- Cercar la informació suficient que em permetés desenvolupar la solució proposada.
- Planificar en el temps les tasques que em caldria dur a terme per a la resolució del problema.
- Documentar les diferents etapes del projecte.
- Implementar el codi que hauria de resoldre el problema plantejat.
- Redactar una memòria ben estructurada que contingués tota la feina feta.
- Exposar el TFC mitjançant una presentació en diapositives de la manera més entenedora possible.

1.3 Enfocament i mètode seguit

El projecte l'he desenvolupat seguint el model de cicle de vida clàssic amb prototipatge. El mètode de desenvolupament del programari que he utilitzat pertany al grup dels anomenats mètodes orientats a l'objecte. En el següent diagrama es representen les successives etapes realitzades fins a l'obtenció del producte final.



Model de cicle de vida clàssic

1.4 Planificació del projecte

El TFC l'he desenvolupat d'acord amb la planificació següent:

Etapa	Dies previstos	Data inici	Data fi
1 ▸ Elaboració del Pla de treball	**19**	**1 març**	**19 març**
Tria de la proposta	2	1 març	2 març
Plantejament i planificació del treball	13	3 març	15 març
Redacció del Pla de treball	4	16 març	19 març
Lliurament PAC1	-	-	19 març
2 ▸ Fonaments i estat de l'art	**17**	**20 març**	**5 abril**
Anàlisi del problema a tractar	5	20 març	24 març
Anàlisi de la solució al problema	6	25 març	30 març
Redacció del Document de l'estat d'art	6	31 març	5 abril
Lliurament PAC2	-	-	5 abril
3 ▸ Disseny	**25**	**6 abril**	**30 abril**

Elaboració del disseny	15	6 abril	20 abril
Redacció del Document de disseny	10	21 abril	30 abril
Lliurament PAC3	-	-	30 abril
4 ▸ Programació	**35**	**1 maig**	**4 juny**
Codificació	31	1 maig	31 maig
Correcció d'errors	4	1 juny	4 juny
5 ▸ Memòria i presentació virtual	**14**	**5 juny**	**18 juny**
Redacció de la memòria	7	5 juny	11 juny
Elaboració de la presentació virtual	7	12 juny	18 juny
Lliurament final	-	-	18 juny
6 ▸ Debat virtual	**?**	**?**	**?**

1.5 Productes obtinguts

Durant el procés d'elaboració del TFC he lliurat, en les dates fixades, els documents que s'indiquen en aquesta taula.

Data lliurament	Document	Contingut del document
19/03/2007	PAC 1 ***Pla de treball***	Objectius del projecte, descripció, planificació temporal, dates de lliurament i eines de programació.
05/04/2007	PAC 2 ***Fonaments i estat de l'art***	Resum del problema a tractar i de la solució proposada.
30/04/2007	PAC 3 ***Disseny***	Actors del sistema, processos de comunicació, interfícies gràfiques, diagrames dels diferents processos, tipus de dades, etc.
18/06/2007	***Lliurament final***	Memòria del projecte, programari i presentació virtual

1.6 Breu descripció dels altres capítols de la memòria

La memòria consta d'aquests capítols: Fonaments i estat de l'art, Disseny, Conclusions, un Glossari, i la Bibliografia, cada un dels quals descriuré breument tot seguit:

Capítol 2: **Fonaments i estat de l'art**

- Descripció de la criptografia de clau simètrica, particularment les xifres de bloc.
- Descripció breu de la teoria sobre la qual es fonamenten els xifradors de bloc: teoria de Shannon.
- Els diferents modes d'operació dels xifradors de bloc: ECB, CBC, CFB, OFB i CTR.
- Notes sobre alguns dels algorismes de xifratge de bloc més coneguts: DES, 3DES, IDEA, Blowfish i AES.
- Alguns comentaris sobre el xifratge d'arxius.
- Exemples de programes comercials per al xifratge d'arxius.
- Descripció del xifrador que he utilitzat per a l'obtenció de la clau de xifratge a partir d'una contrasenya: generador de Geffe.

Capítol 3: **Disseny**

Recull l'anàlisi del problema plantejat - el xifratge d'arxius - i el disseny del programa informàtic que m'ha calgut desenvolupar per tal de resoldre'l.

La descripció de la solució proposada l'he feta en UML. Concretament, aquest capítol conté:

- El corresponent diagrama de casos d'ús.
- El diagrama d'activitats.
- Els diagrames de classes.

El capítol també conté:

- ➲ Imatges de les interfícies gràfiques mitjançant les quals l'usuari interactuarà amb el programa.

En un altre apartat s'explica el procediment que he seguit per tal de construir el generador de Geffe utilitzat per a l'obtenció de les claus criptogràfiques a partir de la contrasenya introduïda per l'usuari.

Capítol 4: **Conclusions**

Resum dels problemes amb els que m'he trobat, aspectes que es podrien millorar del projecte, i una valoració personal del treball fet.

Capítol 5: **Glossari**

Llista dels conceptes que apareixen en aquest document acompanyats del seu significat.

Capítol 6: **Bibliografia**

Llista dels llibres, pàgines web i recursos utilitzats per a la realització del TFC.

2 Fonaments i estat de l'art

2.1 Criptografia de clau simètrica

La criptografia de clau simètrica deu el seu nom al fet que la clau per a xifrar i per a desxifrar missatges és la mateixa.

D'acord amb la criptografia teòrica, la seguretat d'un criptosistema hauria de residir, no en l'algorisme de xifratge, sino en la clau de xifratge, de tal manera que a un atacant li hauria de servir de no res conéixer aquell si desconeix amb quina clau ha estat xifrat el text al que pretén tenir accés. Per tal que la clau amb que ha estat xifrat un text sigui més difícil d'esbrinar, la seva longitud hauria de ser quant més gran millor, doncs en augmentar la longitud de la clau, s'incrementa el nombre total de claus possibles. Si el nombre de bits de la clau que es fa servir en un determinat algorisme de xifratge és *n*, el nombre de claus possibles és igual a *2 elevat a n*.

Per exemple, l'algorisme de xifratge DES usa una clau de *56 bits*, per la qual cosa hi ha *2 elevat a 56* claus possibles. Aquesta quantitat, que l'any de la seva publicació feia de DES un algorisme segur, ha deixat de ser suficient amb la velocitat de càlcul dels ordinadors actuals. En canvi, es podria dir que els algorismes més nous, com ara 3DES, Idea, Blowfish i AES, amb claus de *128 bits* o més, són immunes a atacs de força bruta: per trencar-la es trigaria (amb els ordinadors actuals, repeteixo) gairebé l'edat de l'Univers.

Quan la criptografia de clau simètrica es utilitzada per a la transmissió de dades, sorgeix el problema d'haver d'intercanviar-se la clau l'emissor i el receptor d'una manera segura. En aquest cas, hi ha dos possibles solucions: intercanviar-se la clau a través d'un altre canal, un canal de comunicació alternatiu, diferent a l'utilitzat per a la transmissió del missatge i que sigui segur; o bé, fer servir criptografia de clau pública per a l'intercanvi de la clau a través del mateix canal. Com que aquest projecte tracta del problema del xifratge

d'arxius, que no pas de la transmissió de dades, l'inconvenient de l'intercanvi de la clau de xifratge no l'he tingut pas en compte.

2.2 Xifres de bloc

Un sistema de xifrat per blocs consta, per una part, d'un **xifrador de bloc** (*block cipher*) i, per una altra part, d'un **mode d'operació**.

Un **xifrador de bloc** estarà dissenyat per xifrar un sol bloc. Un bloc és una seqüència de bits d'una certa longitud. La unitat de xifratge d'un criptosistema per blocs és, per definició, el bloc. Una de les característiques pròpies d'un xifrador de bloc serà, doncs, la seva **longitud del bloc**.

Per xifrar un missatge d'una longitud superior a la del bloc, caldrà dividir-lo en blocs, els quals seran subministrats *d'un en un* al xifrador de bloc. Els blocs hi poden ser subministrats tal com resulten de dividir el missatge, sense haver-los d'aplicar cap transformació, o bé es poden sotmetre a una certa operació abans que no siguin processats. És el que s'anomena **mode d'operació**.

Les següents figures il·lustren el xifratge i desxifratge d'un criptosistema per blocs:

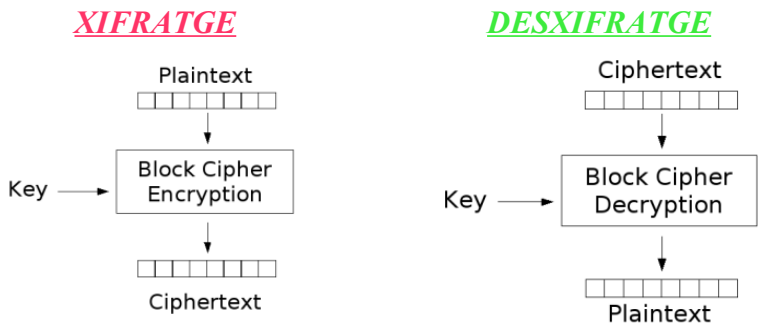


Figura 1: Xifrador de bloc

Una altra de les característiques de les xifres de bloc serà la **longitud de la clau**.

2.2.1 Teoria de Shannon

La **substitució** i la **transposició,** tècniques presents en els criptosistemes actuals, ja havien estat utilitzades en la criptografia clàssica.

Un criptosistema serà més o menys segur en la mesura que els patrons estadístics del text en clar no es detectin també en el text xifrat. En el seu article *Communication Theory of Secrecy Systems*, publicat el 1949, Shannon afirma que la seguretat d'un criptosistema depén bàsicament de la **confusió** i de la **difusió** que aquest és capaç d'introduir en el text xifrat.

Mentre que amb la **confusió** s'incrementa el grau de complexitat de la relació existent entre la clau de xifratge i el text xifrat, amb la **difusió** s'aconsegueix dissipar les propietats estadístiques del text en clar.

En els criptosistemes moderns, els algorismes de xifrat:

- Per una part, generen la confusió mitjançant taules anomenades **taules de substitucions** (S-box).

- Per una altra part, generen la difusió mitjançant les **taules de transposicions o de permutacions** (P-box).

Per aquesta raó, es diu que els algorismes d'aquests sistemes criptogràfics s'ajusten a un **esquema o xarxa de tipus SPN** (substitution-permutation network).

Un cas particular d'esquema SPN és l'anomenat **esquema de Feistel**, amb el qual s'aconsegueix que un mateix algorisme serveixi alhora tant per a xifrar com per a desxifrar. Aquest és l'esquema utilitzat en l'algorisme DES.

Shannon, a més, va introduir el concepte de **producte de xifres**.

2.2.2 Modes d'operació

Els principis de confusió i de difusió de Shannon cal que es compleixin no només a dintre de cada unitat de xifratge. Han de ser-hi també en el text xifrat pres en el seu conjunt.

Ja s'ha dit que en definir un criptosistema per blocs, cal especificar, a més del xifrador de bloc, el seu mode d'operació, és a dir, la manera com els blocs del text en clar hi són entrats per tal que siguin xifrats. La finalitat d'un mode d'operació és estendre la confusió i la difusió a tot el text.

En els següents subapartats es descriuen els cinc modes d'operació més coneguts.

2.2.2.1 ECB (Electronic codebook)

No hi pot haver cap altre mode més elemental que aquest. En ECB els blocs són entrats en el xifrador de bloc sense cap transformació prèvia, és a dir, tal com són en el text en clar. Si el contingut de dos blocs resulta ser el mateix, també ho serà el contingut dels corresponents blocs xifrats, per la qual cosa, un cop fixada la clau de xifratge, el criptosistema actua com una mena de traductor que es limités a fer ús d'un diccionari. El seu nom, de fet, li ve d'això.

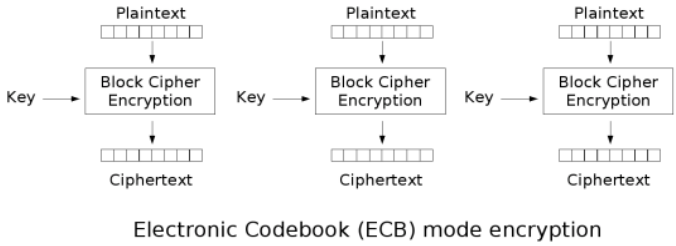


Figura 2: Xifratge ECB

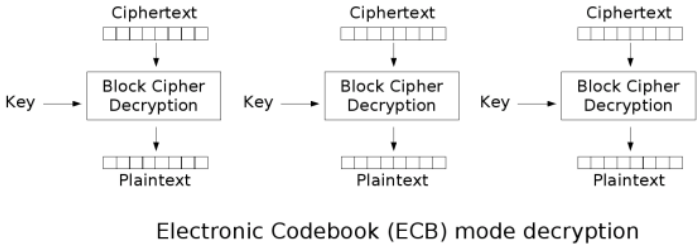


Figura 3: Desxifratge ECB

Sent el més elemental, també és el més vulnerable: la confusió i la difusió que afegeix al text és nul·la. Per tant, en aquest mode i a diferència dels altres, sols existirà confusió i difusió a nivell de bloc, tal com es veu en aquest exemple:

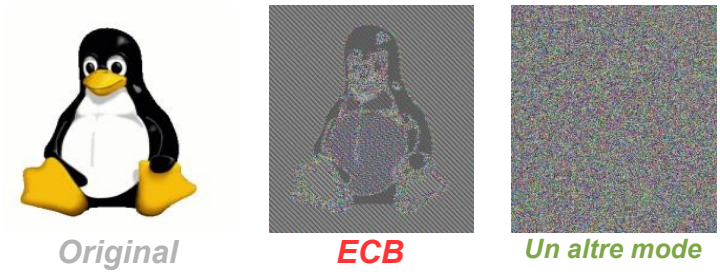


Figura 4: Vulnerabilitat del mode ECB

2.2.2.2 CBC (Cipher-block chaining)

En ordre de complexitat, a l'ECB li segueix el CBC. En aquest mode a cada bloc de text en clar, abans d'entrar-lo al xifrador de bloc, se li aplica l'operació **XOR** amb el bloc xifrat immediatament anterior.

A més, per tal d'obtenir un xifratge únic per a cada missatge, es fa servir un **vector d'inicialització** la longitud del qual és igual a la d'un bloc.

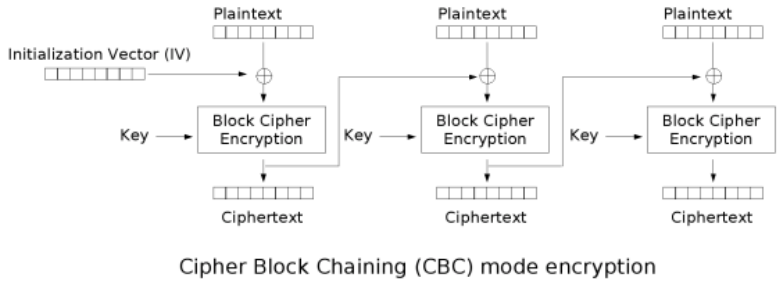


Figura 5: Xifratge CBC

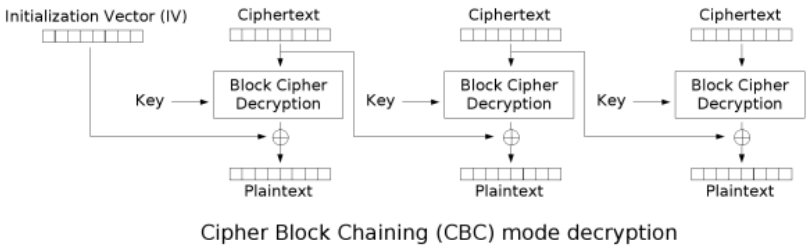


Figura 6: Desxifratge CBC

2.2.2.3 CFB (Cipher feedback)

En CBC els blocs en clar se sumen al resultat del xifratge precedent abans que no entrin en el xifrador de bloc. En CFB, per una part els blocs en clar se sumen directament, sense xifrar, al resultat del xifratge anterior obtenint d'aquesta manera el corresponent bloc xifrat, i per una altra part, aquest últim s'introdueix en el xifrador de bloc abans que no es torni a començar. I així successivament. Aquest mode es pot interpretar com una mena de xifratge de flux on la unitat de xifratge és el bloc i no el bit.

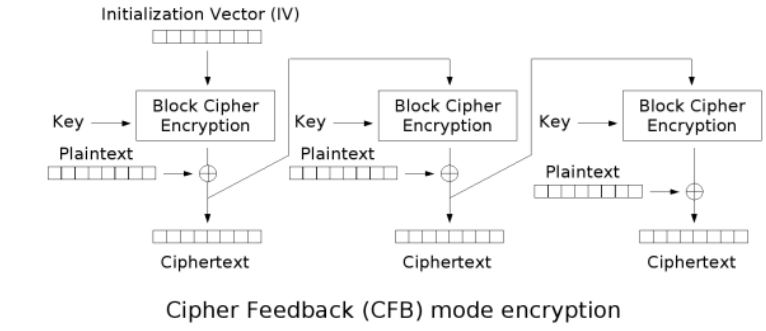


Figura 7: Xifratge CFB

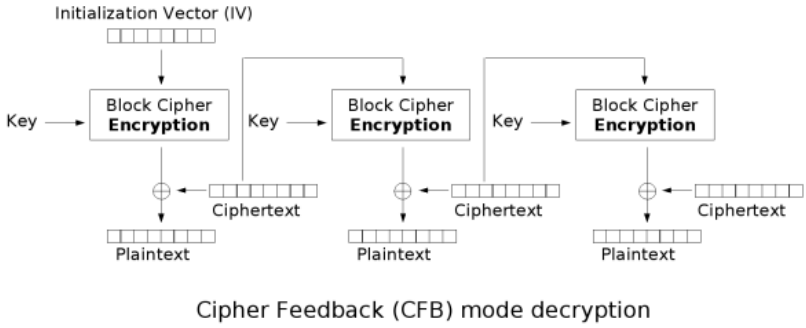


Figura 8: Desxifratge CFB

2.2.2.4 OFB (Output feedback)

Igual que en CFB, en OFB, a diferència de CBC, els blocs en clar no entren directament en el xifrador de bloc. En aquest cas, però, partint del vector d'inicialització i fent-hi un xifratge recursiu s'obté un flux de blocs, els quals es van sumant als blocs en clar per tal d'obtenir els blocs xifrats.

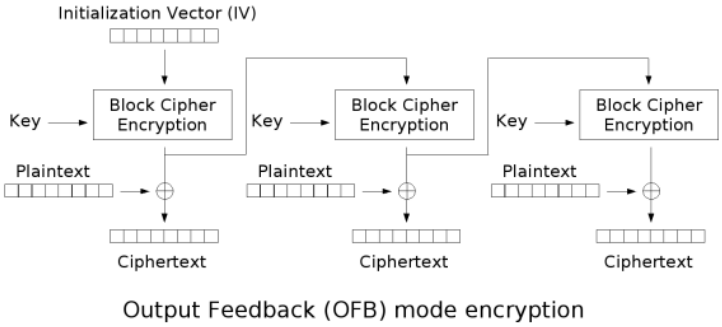


Figura 9: Xifratge OFB

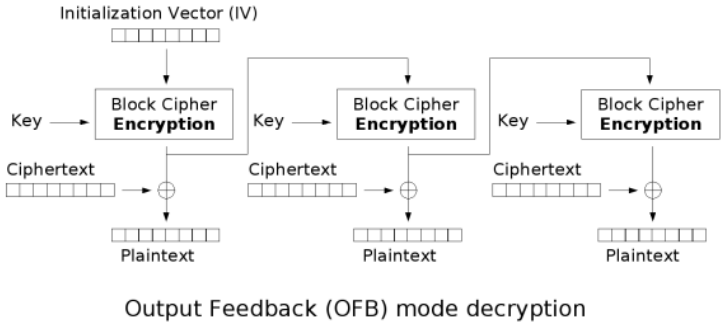


Figura 10: Desxifratge OFB

2.2.2.5 CTR (Counter)

L'única diferència de CTR respecte de OFB és que el flux de blocs que caldrà sumar als blocs en clar, no s'obté amb un xifratge recursiu del xifrador de bloc, començant amb el vector d'inicialització, sino mitjançant una funció comptador.

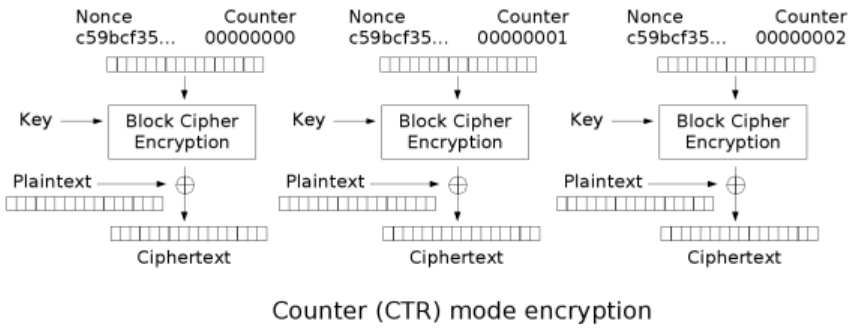


Figura 11: Xifratge CTR

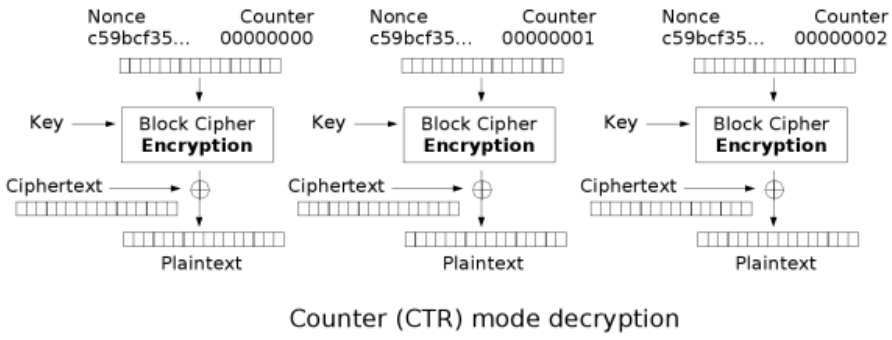


Figura 12: Desxifratge CTR

2.2.3 Algorismes de xifratge

En aquest apartat comentaré els xifradors de bloc més coneguts.

2.2.3.1 DES (Data Encryption Standard)

DES va ser aprobat el 1976 com a estàndard federal pel NBS (National Bureau of Standards) - ara, NIST (National Institute of Standards and Technology). Aquest algorisme xifra blocs de 64 bits i utilitza claus d'aquesta mateixa longitud. La seva estructura s'ajusta a l'esquema de Feistel representat en la figura següent. El bloc en clar de 64 bits es divideix en dues meitats de 32 bits, les quals són processades alternativament, en forma creuada.

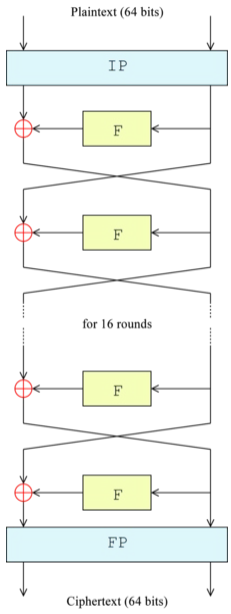


Figura 13: Xifrador DES

Aquí, *F* representa la **funció de Feistel.** *IP* i *FP* són dues permutacions, l'una inversa de l'altra, les quals no són criptogràficament significatives (en l'època en que va ser dissenyat l'esquema, la seva utilitat fou la de facilitar la càrrega i

descàrrega de blocs sobre maquinari). Entre la permutació inicial i la final hi ha 16 iteracions.

Pel que fa a la funció de Feistel, rep com a entrada mig bloc de 32 bits i una subclau de 48 bits. Consta de 4 operacions:

1. **Expansió**: el mig bloc de 32 bits s'expandeix a 48 bits mitjançant la funció *E*.
2. **XOR**: els 48 bits de l'expansió se sumen als 48 bits d'una subclau. Per a cada iteració de l'esquema de Feistel hi haurà una subclau distinta. Aquestes 16 subclaus s'obtenen a partir de la clau de xifratge de 64 bits mitjançant un generador de subclaus.
3. **Substitució**: els 48 bits de la suma anterior es divideixen en 8 miniblocs de 6 bits. Aquests miniblocs són processats per 8 taules *S1, S2 ... S8* de subtitucions o S-box. La sortida de cada S-box és un altre minibloc de 4 bits. En resultarà, donc, un total de 32 bits.
4. **Permutació**: aquest bloc de 32 bits és processat per una taula *P* de permutacions o P-box.

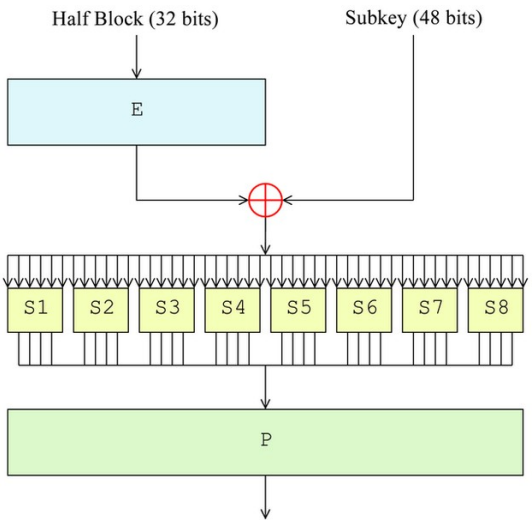


Figura 14: Funció Feistel del DES

La generació de les 16 subclaus de 48 bits a partir de la clau de xifratge de 64 bits es fa de la següent manera:

Mitjançant l'elecció permutada PC1 se seleccionen 56 bits dels 64 inicials i aquests 56 bits es divideixen en 2 meitats de 28 bits. A continuació, es fan 16 iteracions d'un mateix procés consistent cadascuna en desplaçar un bit o dos cap a l'esquerra, segons la iteració, cada una d'aquestes dues meitats, i en seleccionar, mitjançant l'elecció permutada PC2, 24 bits de cada meitat a fi d'obtenir una subclau de 48 bits.

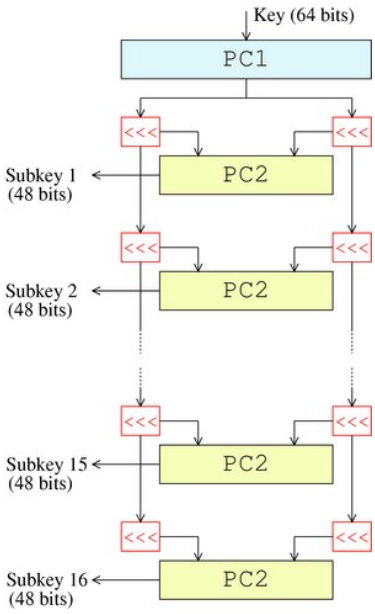


Figura 15: Generació de subclaus en DES

Tal com ja he apuntat en un apartat anterior, l'esquema de Feistel té l'avantatge de que un mateix algorisme serveix alhora tant per a xifrar com per a desxifrar, la qual cosa simplifica la implementació, especialment si s'ha de fer sobre maquinari. L'única diferència entre el xifratge i el desxifratge es troba en la generació de les subclaus: per al primer els bits s'han de desplaçar cap a l'esquerra, mentres que per al desxifratge ho han de fer cap a la dreta.

	DES (Data Encryption Standard)
Autor	IBM
Data de publicació	1975 (estandaritzat en gener del 1977)
Algorisme pare	Lucifer
Algorismes fills	Triple DES, G-DES, DES-X, LOKI89, ICE
Longitud de la clau	56 bits
Longitud del bloc	64 bits
Nombre d'iteracions	16
Comentaris	Actualment es considera insegur davant d'atacs per força bruta

2.2.3.2 3DES (Triple DES)

En 1998, un atac per força bruta (provant una per una cada possible clau) va aconseguir trencar una clau DES en menys de 24 hores, la qual cosa va plantejar la necessitat d'haver de substituir DES per un altre estàndard que fes servir una clau més llarga. El nou estàndard seleccionat va ser l'AES. No obstant això, molts dels usuaris de DES utilitzen ara el 3DES, per exemple la majoria de les targetes de crèdit i altres mitjanç de pagament electrònic. L'algorisme 3DES consisteix en aplicar DES tres vegades consecutives amb claus diferents en cada una.

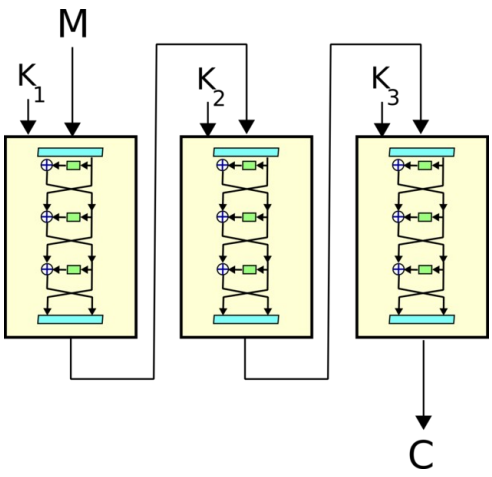


Figura 16: Xifrador 3DES

El fet que es tripliquen el nombre d'operacions implica que 3DES sigui molt més lent que el seu predecessor DES.

	3DES (Triple DES)
Autor	IBM
Data de publicació	1978
Algorisme pare	DES
Algorismes fills	-
Longitud de la clau	112 (2TDES) or 168 bits (3TDES)
Longitud del bloc	64 bits
Nombre d'iteracions	48

2.2.3.3 IDEA (International Data Encryption Algorithm)

IDEA va ser una de les propostes per substituir DES. Aquest algorisme opera amb blocs de 64 bits i fa servir una clau de 128 bits. Consisteix en 8 iteracions i una transformació de sortida (equivalent a mitja iteració). El xifratge i el desxifratge són semblant. En IDEA la seguretat s'aconsegueix combinant distints grups "incompatibles" (s'enten que en un cert sentit algebraic): grup de suma modular, grup de multiplicació modular i grup XOR.

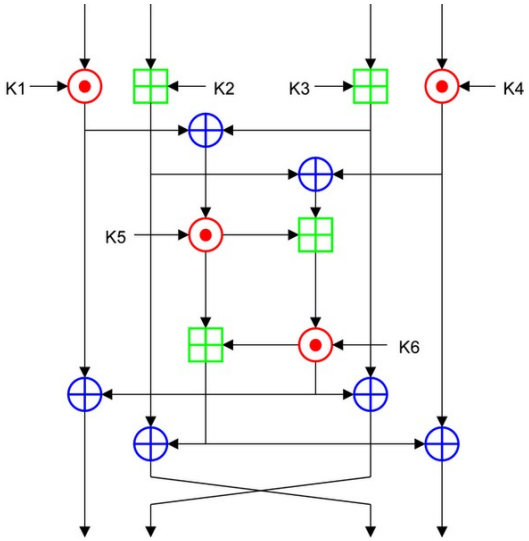


Figura 17: Xifrador IDEA

Respecte del DES, d'aquest algorisme en destaquen aquests punts:

- Espai de claus molt més gran: 2 elevat a 128.
- Totes les operacions són algebraiques.
- No hi ha operacions de desplaçaments de bits, la qual cosa facilita la programació en alt nivell.
- Esquema més eficient que el de Feistel, doncs en cada iteració es modifiquen tots els bits de bloc i no només la meitat.
- Es pot utilitzar amb tots els modes d'operació definits per al DES.

	IDEA
Autors	James Massey, Xuejia Lai
Data de publicació	1991
Algorisme pare	PES
Algorismes fills	MMB, MESH, Akelarre, IDEA NXT (FOX)
Longitud de la clau	128 bits
Longitud del bloc	64 bits
Nombre d'iteracions	8.5

2.2.3.4 Blowfish

Blowfish va ser una altra proposta per subtituir a DES, però a diferència de IDEA conserva l'esquema de Feistel. El seu dissenyador, a més, no el va voler patentar.

La longitud del bloc d'aquest algorisme és de 64 bits i la clau pot anar dels 32 als 448 bits. Tal com s'observa en la figura, conté 16 taules de permutacions P1, P2 ... P16, cada una de les quals opera en una iteració, i dues taules més, P17 i P18, que se situen al final de tot el procés.

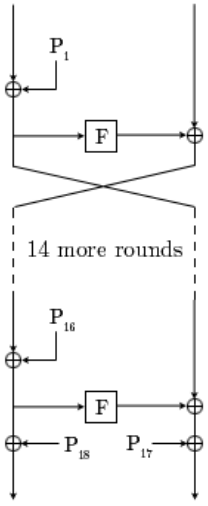


Figura 18: Xifrador Blowfish

La funció de Feistel divideix les entrades de 32 bits en 4 miniblocs de 8 bits i sobre cada un d'aquests miniblocs opera una de les 4 taules de substitucions S1, S2, S3 i S4 que defineixen la funció.

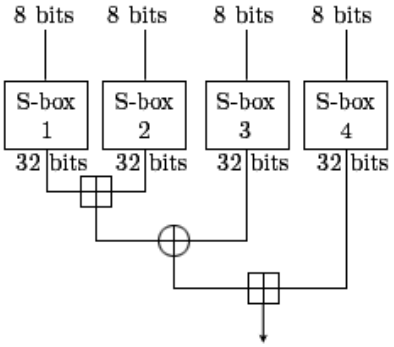


Figura 19: Funció Feistel de Blowfish

La clau de xifratge s'utilitza per a inicialitzar les taules de permutacions i de substitucions segons un procediment no exempt d'una certa complexitat.

	Blowfish
Autor	Bruce Schneier
Data de publicació	1993
Algorisme pare	-
Algorismes fills	Twofish
Longitud de la clau	32, 40, 48, 56 ... 448 bits ; per defecte es de 128 bits
Longitud del bloc	64 bits
Nombre d'iteracions	16

2.2.3.5 AES (Advanced Encryption Standard)

En 1997, el NIST va convocar un concurs amb la finalitat d'escollir el nou criptosistema destinat a subtituir el fins aleshores estàndar DES. Després d'un llarg procés, dels 15 algorismes admesos, el guanyador va ser l'algorisme Rijndael (nom que s'obté de fusionar els noms dels seus inventors). El 26 de maig del 2002 es va convertir en l'estàndard oficial i a partir de llavors es coneix amb el nom de AES.

Xifra blocs de 128 bits i utilitza claus de 128, 192 o 256 bits. AES opera sobre una matriu de 4x4 bytes anomenada **matriu d'estat**. Per al xifrat cada iteració, llevada l'última, consta de 4 etapes: SubBytes, ShiftRows, MixColumns i AddRoundKey. La iteració final substitueix la fase MixColumns per una altra instància de AddRoundKey.

Etapa SubBytes

Cada byte de la matriu d'estat en clar és substituït conforme a una taula de subtitució.

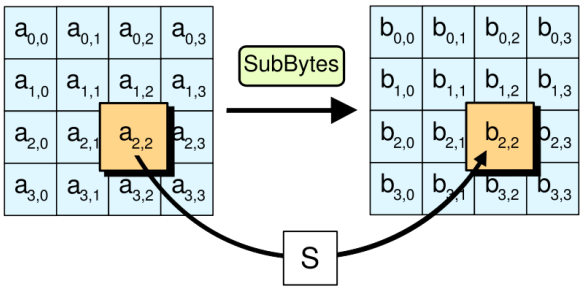


Figura 20: Etapa SubBytes de AES

Etapa ShiftRows

Els 4 bytes de cada fila de la matriu d'estat experimenten una rotació cíclica d'acord amb un determinat offset.

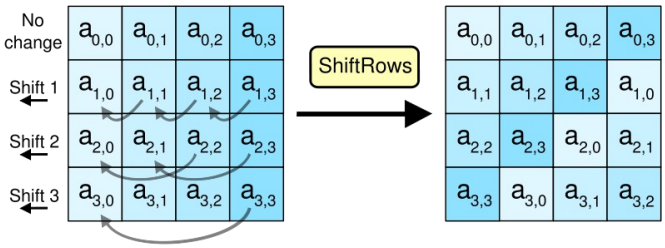


Figura 21: Etapa ShiftRows de AES

Etapa MixColumns

Els 4 bytes de cada columna es combinen usant una transformació lineal invertible.

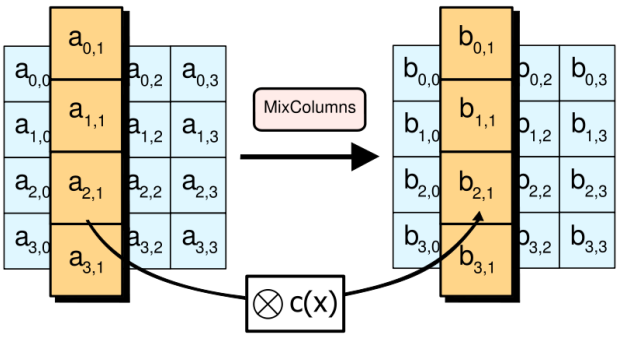


Figura 22: Etapa MixColumns de AES

Etapa AddRoundKey

A cada iteració s'obté una subclau de la clau principal, usant la iteració de la clau; cada subclau és una matriu de la mateixa dimensió que la matriu d'estat. En aquesta etapa, cada byte de la matriu d'estat es combina amb un XOR amb el byte corresponent de la subclau.

De moment, ningú no ha estat capaç de dur a terme un atac amb éxit contra AES.

	AES (Advanced Encryption Standard)
Autors	Vincent Rijmen, Joan Daemen
Data de publicació	1998 (estandaritzat el maig del 2002)
Algorisme pare	Square
Algorismes fills	Anubis, Grand Cru
Longitud de la clau	128, 192 or 256 bits
Longitud del bloc	128 bits
Nombre d'iteracions	10, 12 or 14 (depenent de la longitud de la clau)

2.3 Emmagatzematge d'arxius xifrats

D'alguna manera, la criptografia és l'art de convertir un secret gran en un altre de petit. Un fitxer s'encripta perque es vol guardar en secret. En encriptar-lo, el secret del fitxer s'haurà transformat en el secret de la clau de xifratge.

Mentres que en la criptografia per a l'intercanvi de fitxers la durada del secret de la clau és igual al temps que trigui l'enviament, en el xifratge d'arxius el secret de la clau haurà de perdurar fins que l'arxiu no abandoni la seva condició de secret o no sigui destruït.

En la pràctica no té sentit emmagatzemar un arxiu xifrat i conservar al mateix temps l'original en clar, de manera permanent. Un cop xifrat aquest últim, cal

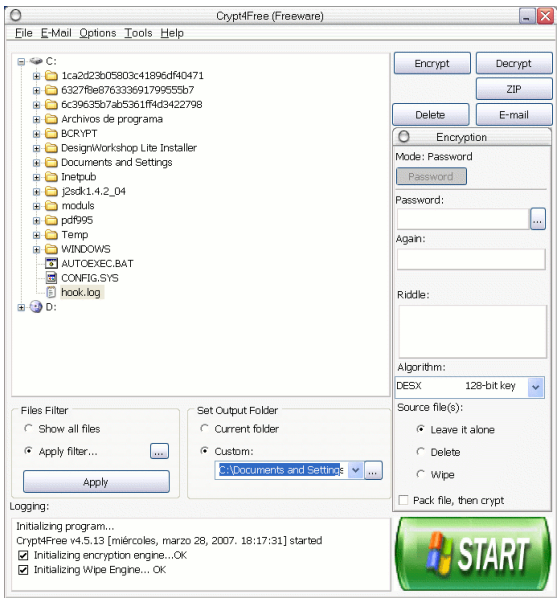
que sigui eliminat per tal que l'arxiu es mantingui guardat realment en secret. La clau de xifratge en el cas de l'emmagatzematge d'arxius xifrats, doncs, torna a cobrar una importància que no la té pas en el cas de l'intercanvi d'arxius xifrats. Suposant que l'arxiu en clar hagués estat efectivament esborrat, si es perdés la clau amb que havia estat emmagatzemat, es perdria de fet l'arxiu, doncs no existiria cap mecanisme que permetés desencriptar-lo. En canvi, l'original en clar d'un fitxer que per a transmetre'l ha estat encriptat no s'elimina. En aquest cas la pérdua de la clau de xifratge no implica la pérdua del fitxer i el problema es resol amb una nova retransmissió segura.

A diferència del xifratge per a l'intercanvi d'arxius, en la criptografia per al emmagatzematge segur la velocitat de l'algorisme de xifratge/desxifratge no és un factor crític. En el darrer cas, el sincronisme que hi ha d'haver entre l'emissor de l'arxiu xifrat i el receptor és una condició aliena al problema.

Convé senyalar, finalment, que els algorismes emprats per xifrar fitxers plans no són adients per encriptar bases de dades. En aquest cas, la problemàtica associada és força més complicada i els criptosistemes requerits són molt més complexos.

2.4 Productes

De la llarga llista de productes per al xifratge d'arxius que es poden descarregar d'Internet, n'he seleccionat uns quants. Hi ha, en tots aquests exemples, una versió lliure o una de prova, vàlida fins a un cert nombre de dies.



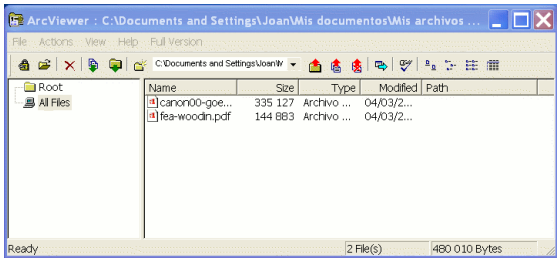
Crypt4Free

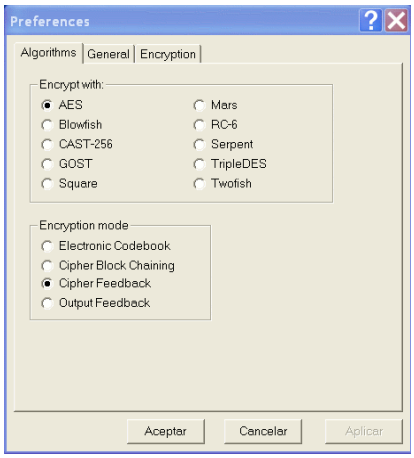
Permet xifrar amb DESX i Blowfish.

Es tracta de la versió lliure de **AEP 2007 PRO**, la qual conté, a més dels dos anteriors: AES 256 bits, MARS, Serpent, Cast ... fins a un total de 18 algorismes criptogràfics.

Descàrrega:

http://www.secureaction.com





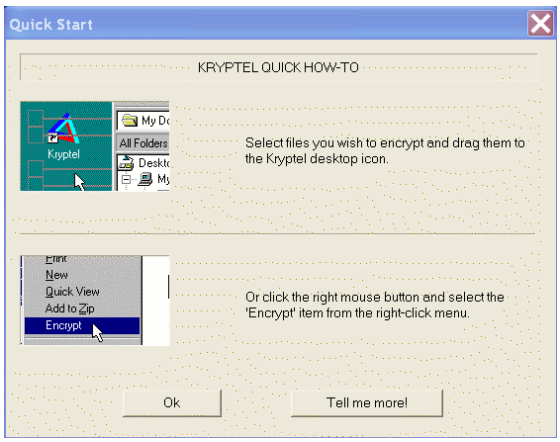
FineCrypt Archiver

Permet xifrar amb AES 256 bits, Blowfish, CAST-256, GOST, Square, MARS, RC-6, Serpent, 3D i Twofish.

Admet també tots els modes d'operació descrits en un apartat anterior: ECB, CBC, CFB i OFB.

Descàrrega:

http://crypto-systems.com/



Krypten Lite

Permet xifrar amb DES i mode ECB.

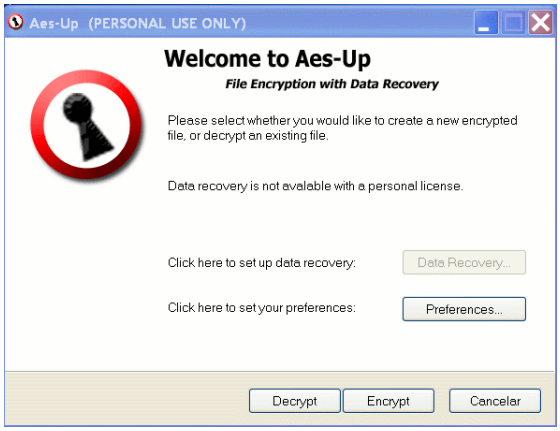
Es tracta de la versió lliure de **Krypten Full**, la qual, a més, té AES, 3DES, Blowfish, Twofish, Serpent i IDEA.

Amb la versió completa es poden encriptar carpetes.

Descàrrega:

http://www.kryptel.com/

A diferència dels anteriors, aquests altres productes només implementen un determinat algorisme. Es tracta, concretament, de dos xifradors AES, un xifrador Blowfish i un xifrador IDEA.



Aes-Up

Xifra amb AES 256 bits.

Conté, a més, un generador de passwords.

Descàrrega:

http://www.axiom-infosec.com/

dsCrypt

Aquest és un altre xifrador en AES 256 bits i mode CBC.

Descàrrega:

http://members.ozemail.com.au/~nulifetv/freezip/freeware/

Emsa EZ Encryption Tool

Implementa Blowfish. Permet xifrar tant fitxers com text pla.

Descàrrega:

http://www.e-systems.ro

Des del prompt del DOS s'executa:

```
IDEA +/- <File>
```

El primer paràmetre especifica el mode: + per xifrar; - per desxifrar.

Per exemple:

```
IDEA + a:\mailbox.txt
```

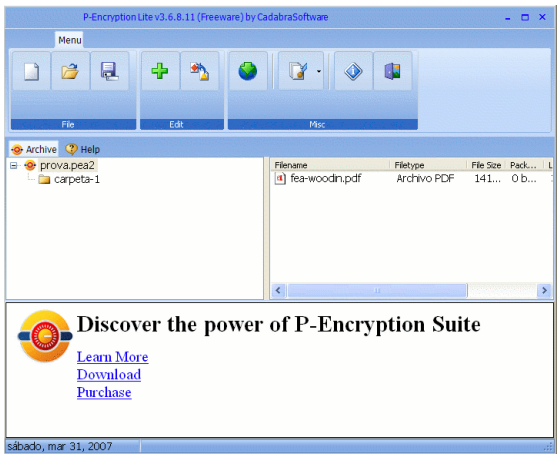
IDEA program

Xifrador IDEA per a DOS.

Descàrrega:

http://cypherspace.org/adam/rsa/idea.html

Els tres productes que referencio a continuació es podrien anomenar **valisses**: permeten guardar arxius a dintre d'una carpeta que es tanca criptogràficament amb una clau.

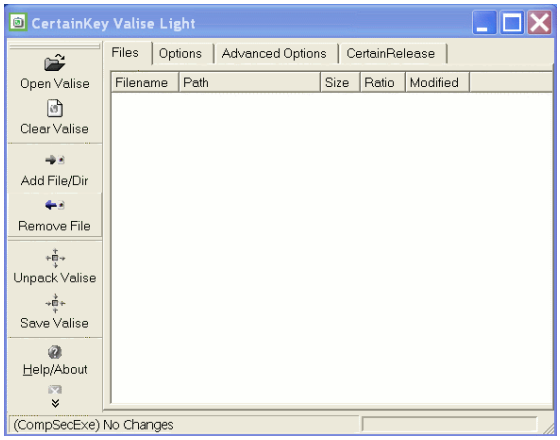


P-Encryption

Utilitza AES, Blowfish, Twofish i 3DES.

Descàrrega:

http://www.cadabrasoftware.com/



CertainKey Valise Light

La versió lliure només utilitza DES. El programa complet fa servir, a més: Serpent, Twofish, AES i 3DES.

Descàrrega:

http://www.certainkey.com/demos/

Folder Lock

Permet codificar arxius amb Blowfish 256 bits.

Descàrrega:

http://www.newsoftwares.net/folderlock/

Finalment, hi he afegit tres productes que, tot i plantejar una estratègia diferent a la que implementaré en el meu projecte per a l'emmagatzematge secret d'arxius, representen una altra solució que cal tenir en compte. Aquestes tres aplicacions creen una **unitat de disc virtual** susceptible de ser bloquejada amb una contrasenya.

CrossCrypt	→	http://www.scherrer.cc/crypt/
Cryptainer LE	→	http://www.cypherix.com/
BestCrypt	→	http://www.jetico.com/

2.5 Generador de Geffe

Les xifres dels criptosistemes de clau simètrica es classifiquen en xifres de bloc i xifres de flux. El problema del xifratge d'arxius es resol amb algorismes de xifratge de bloc com els descrits en apartats anteriors. Aquests algorismes fan servir claus de xifratge la longitud de les quals pot ser de 64 bits (DES), de 128 o més bits (3DES, IDEA, Blowfish i AES). En aquest punt, les **xifres de flux** poden representar un mecanisme òptim per a la generació d'aquestes claus.

En general, els criptosistemes de xifratge de flux més elementals s'implementen mitjançant registres de desplaçament realimentats linealment o **LFSR** (Linear feedback shift register):

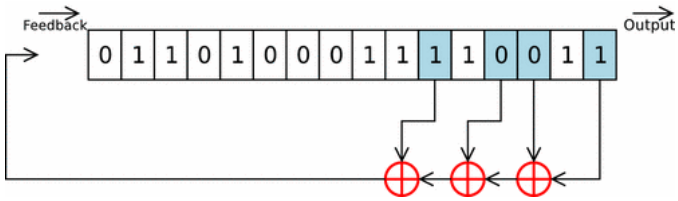


Figura 23: Exemple d'un LFSR

Un LFSR de longitud n és un dispositiu format per n cel·les de memòria i n portes lògiques (en la figura només hi ha representades les portes lògiques ON; les portes OFF no hi són). L'**estat del registre** en l'instant *t* de temps és el conjunt de valors binaris continguts en les seves cel·les en l'instant *t*. A cada impuls de rellotge els bits es desplacen una cel·la en un mateix sentit i es fa la corresponent suma binaria associada al disseny propi del dispositiu.

S'anomena estat absorbent d'un LFSR a aquell en el qual totes les seves cel·les prenen el valor nul. Si un LFSR cau en l'estat absorbent, s'atura, ja que una seqüència de tot zeros produirà una altra seqüència de zeros i el LFSR es perpetuarà per sempre més en aquest estat. Per tant, si no es té en compte l'estat absorbent, el nombre màxim d'estats que pot tenir un LFSR de longitud *n* és igual a $2^n - 1$.

En els casos que un LFSR mai acabi en un estat absorbent, els seus estats recorreran una seqüència cíclica d'un cert període. El període màxim que pot tenir un LFSR de longitud *n* és igual al seu nombre màxim d'estats, és a dir, $2^n - 1$. El fet que un LFSR pugui arribar a tenir el període màxim només depen del seu disseny, és a dir, independentment de quin sigui el seu estat inicial, el seu període sempre serà el període màxim. Obviament, els LFSRs més segurs són els de període màxim, i dintre d'aquest grup, el seu grau de seguretat és una funció creixent del període: el registre serà tant més segur quant més llarg sigui el seu període màxim. Augmentar la seguretat tractant de trobar LFSRs cada cop més segurs resulta ser, en la pràctica, una opció materialment gens viable.

Una altra solució al problema plantejat de com incrementar la seguretat dels criptosistemes de flux és la dels generadors no lineals o **NLFSR** (Non linear feedback shift register), l'esquema general dels quals es representa en la següent figura.

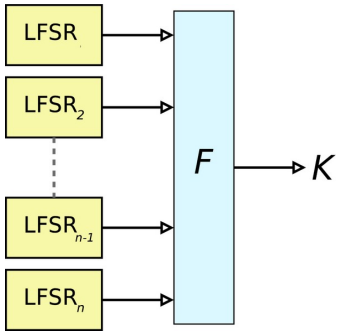


Figura 24: Xifrador NLFSR

La funció F és no lineal i rep com a entrada la sortida d'un cert nombre de LFSRs.

Exemples d'aquest tipus de criptosistemes de flux no lineals són: el generador de Geffe, el generador de Beth-Piper i el generador multivelocitat de Massey-Rueppel.

En l'aplicació del projecte que estic desenvolupant implementaré un generador de Geffe per a generar, a partir d'una contrasenya, la clau criptogràfica requerida per l'algorisme de bloc en el xifratge d'arxius.

Un generador de Geffe es construeix amb 3 LFSR, dos dels quals generen dues seqüències, i el tercer determina la funció de sortida.

A cada impuls t del rellotge tenim dos valors de sortida $s_1(t)$ i $s_2(t)$ dels registres $LFSR_1$ i $LFSR_2$, respectivament. El valor $s_0(t)$ del $LFSR_0$ determina el valor final de sortida s(t) del generador:

$$s(t) = s_1(t) \quad \text{si} \quad s_0(t) = 0$$

i

$$s(t) = s_2(t) \quad \text{si} \quad s_0(t) = 1.$$

3 Disseny

3.1 Diagrama de casos d'ús

El problema plantejat en aquest projecte consta d'un únic cas d'ús: xifrar i desxifrar arxius. Tal com s'observa en el diagrama, inclou la possibilitat de que l'usuari seleccioni l'algorisme de xifratge, seleccioni l'arxiu o la carpeta que vol xifrar o desxifrar, i en el cas de xifrar un arxiu o una carpeta se li demanarà si vol esborrar l'arxiu o la carpeta en clar.

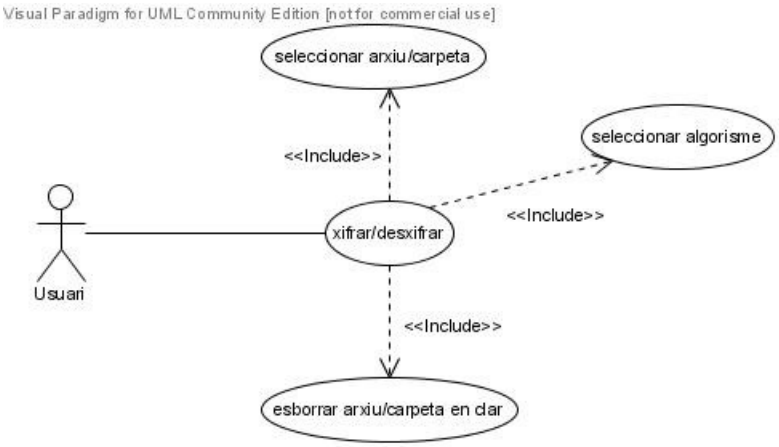


Figura 25: Diagrama de casos d'ús

Qualsevol usuari autoritzat per a executar el programa podrà xifrar els arxius i les carpetes sobre els que tingui drets de lectura. Per desxifrar un arxiu o una carpeta caldrà, a més, que conegui la contrasenya amb la què va ser xifrat.

3.2 Diagrama d'activitats

Del següent diagrama es dedueix que per a xifrar o desxifrar un arxiu l'usuari haurà de seguir un esquema clarament seqüencial. Des de qualsevol punt es podrà cancel·lar el procés i tornar a l'inici.

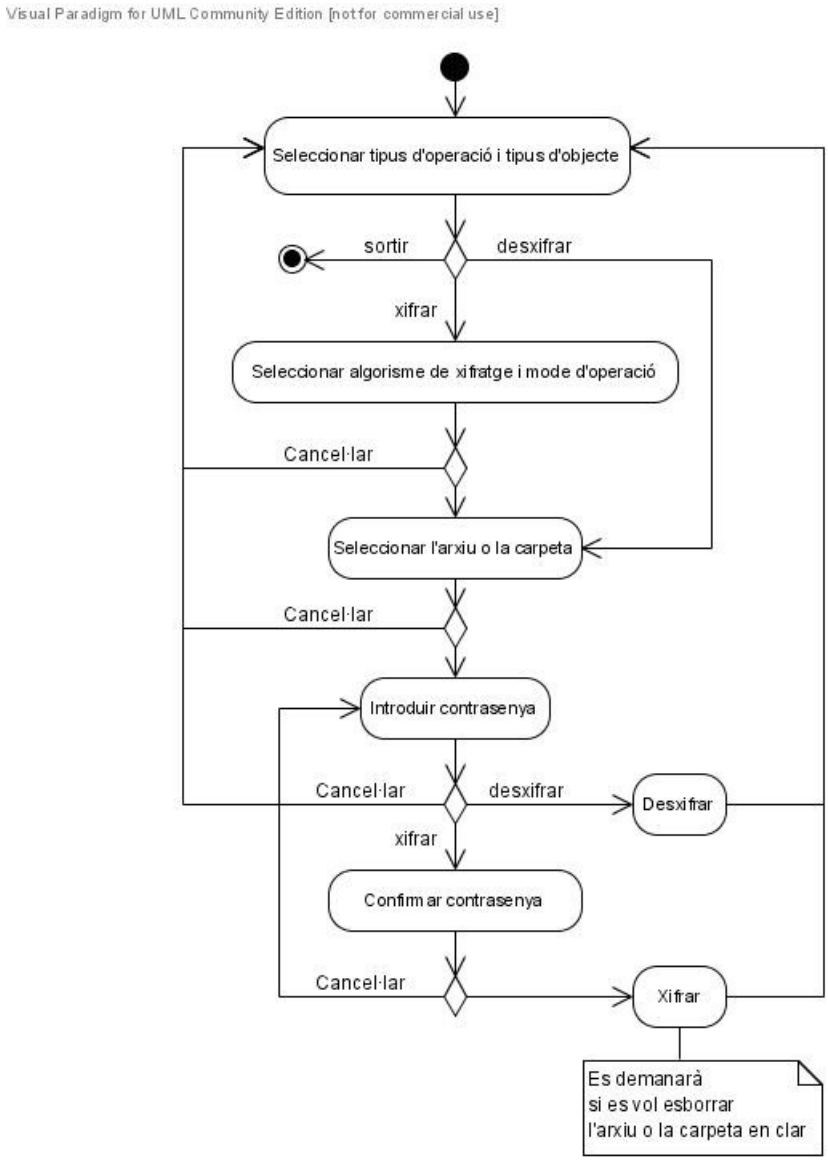


Figura 26: Diagrama d'activitats

Així doncs, si l'usuari vol **xifrar** un arxiu o una carpeta:

1) Seleccionarà el tipus d'operació **xifrar** i el tipus d'objecte (arxiu o carpeta).

2) L'usuari seleccionarà l'algorisme de xifratge (DES, 3DES, IDEA, Blowfish, AES) i el mode d'operació (ECB, CBC, CFB, OFB).

3) Després seleccionarà l'arxiu o la carpeta que vol xifrar.

4) Introduirà la contrasenya amb la que es vol xifrar l'arxiu.

5) Caldrà repetir-la per tal de confirmar que no s'ha equivocat en teclejar-la.

6) Llavors, es xifrarà l'arxiu o la carpeta.

7) Un cop xifrat se li demanarà a l'usuari si vol esborrar l'arxiu o la carpeta original en clar.

Per **desxifrar** un arxiu o una carpeta xifrada anteriorment:

1) L'usuari seleccionarà el tipus d'operació **desxifrar.** En aquest cas la selecció del tipus d'objecte, arxiu o carpeta, és indiferent.

2) Tot seguit seleccionarà l'arxiu xifrat que vol desxifrar.

3) Introduirà la contrasenya amb la que va ser xifrat l'arxiu o la carpeta.

4) Llavors, es desxifrarà l'arxiu o la carpeta.

Internament, l'operació de **xifratge** consistirà en aquesta seqüència d'operacions:

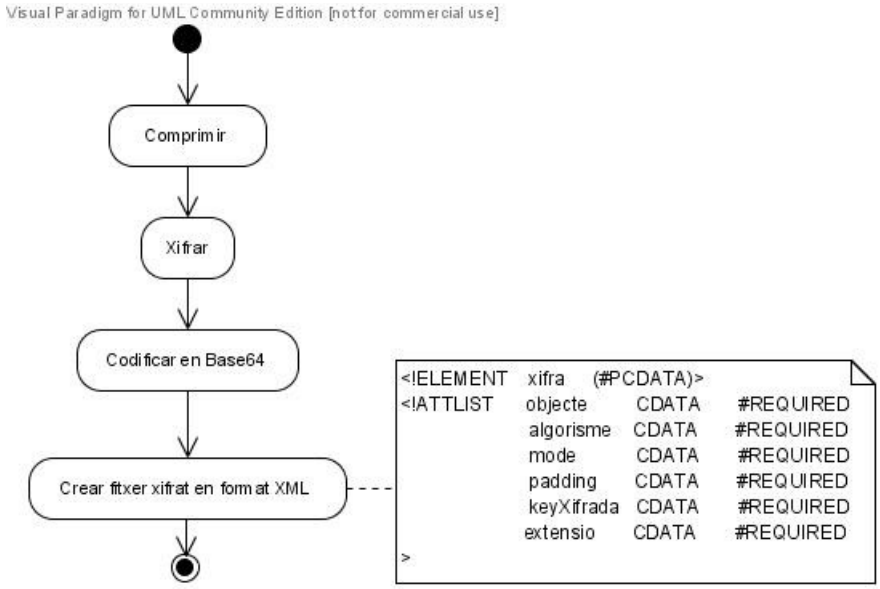


Figura 27: Diagrama de l'operació de xifratge

Abans de xifrar-lo amb l'algorisme i el mode d'operació, l'arxiu (o la carpeta) escollit serà comprimit en format zip. Un cop xifrat serà codificat en Base64, i finalment es crearà un fitxer en format XML que contindrà la xifra amb els atributs: objecte (arxiu o carpeta), algorisme, mode, padding, keyXifrada i extensió. Aquests atributs seràn utilitzats en el procés de desxifratge de l'arxiu. La clau xifrada serà la clau criptogràfica generada pel generador Geffe a partir de la contrasenya introduïda però xifrada amb el mateix algorisme, mode, padding i clau utilitzats per xifrar l'arxiu. També haurà estat codificada en Base64.

En el cas de desxifrar un arxiu la seqüència d'operacions serà la inversa de la del xifratge:

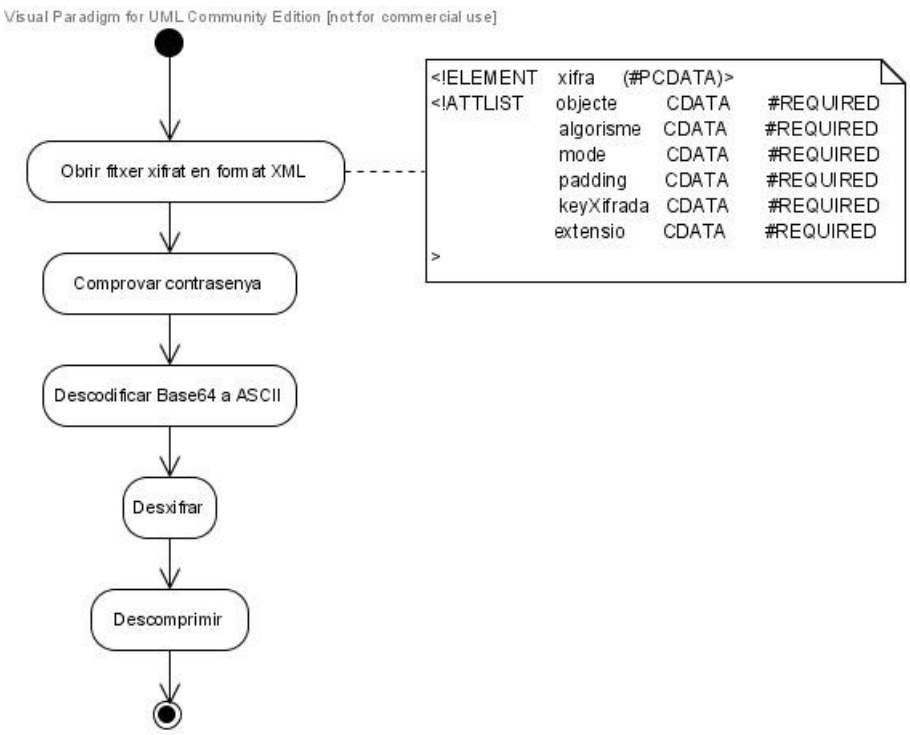


Figura 28: Diagrama de l'operació de desxifratge

Un cop obert el fitxer XML resultat del xifratge, es comprovarà que la contrasenya introduïda coincideix amb la utilitzada per xifrar l'arxiu, per a la qual cosa la xifrarà amb l'algorisme, el mode, el padding i la clau criptogràfica generada pel generador Geffer, i comparará el seu codi en Base64 amb el valor de l'atribut keyXifrada del fitxer XML. Si la contrasenya és vàlida, llavors desxifrarà la xifra i es descomprimirà per a obtenir l'arxiu o la carpeta en clar.

3.3 Interfícies gràfiques

La interacció de l'usuari amb el programa de xifratge d'arxius tindrà lloc mitjançant interfícies gràfiques. Respectant el flux de control del diagrama d'activitats de l'apartat anterior, les interfícies d'usuari, en ordre d'aparició, són les següents:

1. Des de la pantalla inicial l'usuari podrà seleccionar el tipus d'operació que vol efectuar i el tipus d'objecte sobre el que vol operar. També des d'aquí tancarà el programa.

Figura 29: GUI Pantalla Inicial

2. El següent quadre de diàleg li permetrà seleccionar l'algorisme de xifratge i el mode d'operació.

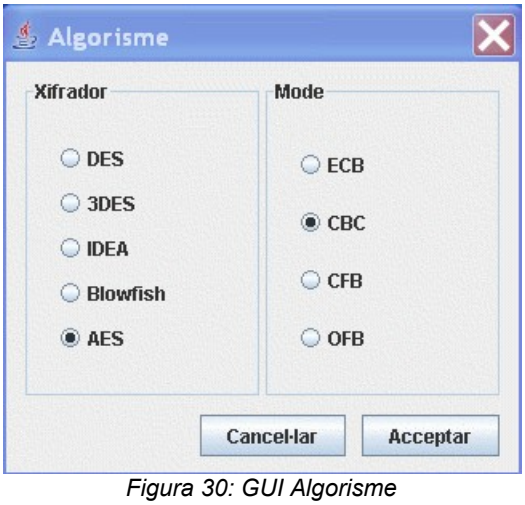


Figura 30: GUI Algorisme

3. Amb un explorador l'usuari podrà seleccionar l'arxiu o la carpeta sobre la que vol operar.

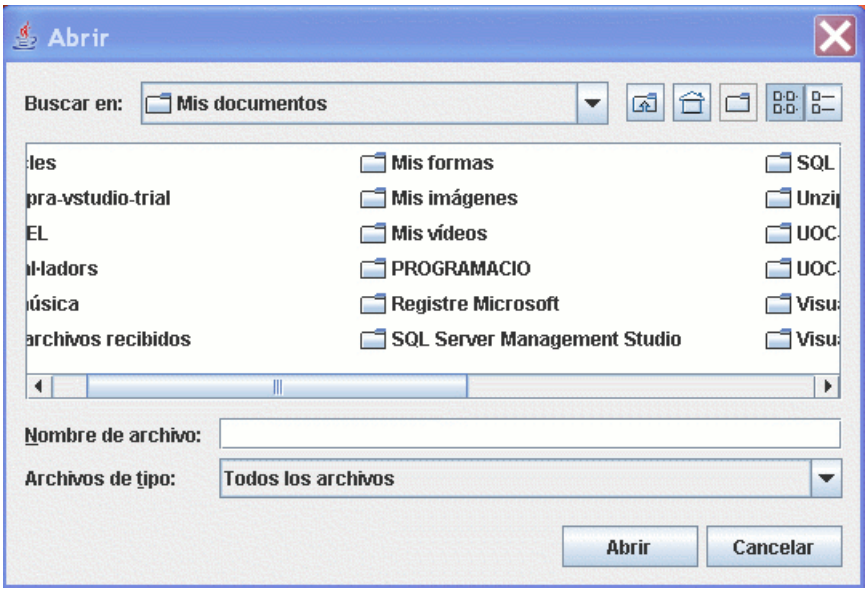


Figura 31: GUI Explorador Sistema de fitxers

4. A través d'un altre quadre de diàleg l'usuari introduirà la contrasenya de l'arxiu o la carpeta que vol xifrar, o la contrasenya amb la que va ser xifrat si es tracta de desxifrar-lo.

Figura 32: GUI Contrasenya

5. En el cas de xifrar un arxiu o una carpeta, caldrà confirmar la contrasenya.

Figura 33: GUI Confirmar contrasenya

6. El següent quadre de diàleg informarà a l'usuari que el procés de xifratge o de desxifratge ha finalitzat.

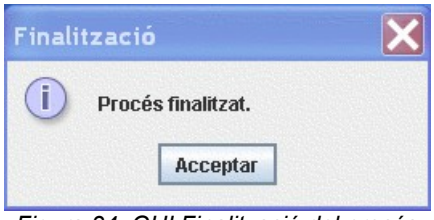


Figura 34: GUI Finalització del procés

7. En el cas d'un xifratge, se li demanarà si vol esborrar l'arxiu o la carpeta original en clar.

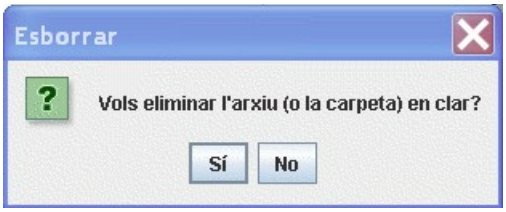


Figura 35: GUI Esborrar arxiu

Arribat a aquest punt, l'usuari podrà tancar el programa o continuar xifrant i desxifrant arxius des de la pantalla inicial.

3.4 Descripció del generador de la clau

La clau criptogràfica s'obtindrà a partir de la contrasenya introduïda per l'usuari fent servir un generador de Geffe. En un document anterior vaig explicar que un generador de Geffe és un xifrador de flux no lineal construït amb 3 LFSR (Linear feedback shift register), dos dels quals generen dues seqüències i el tercer determina la funció de sortida.

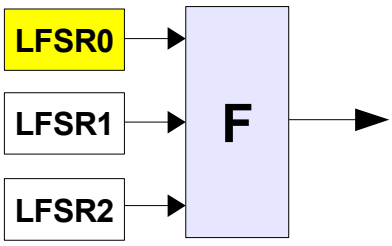


Figura 36: Diagrama generador Geffe

A cada impuls t de rellotge tenim dos valors de sortida $s_1(t)$ i $s_2(t)$ dels registres LFSR1 i LFSR2, respectivament. El valor $s_0(t)$ del LFSR0 determina el valor final de sortida s(t) del generador:

$$s(t) = s_1(t) \quad si \quad s_0(t) = 0$$

i

$$s(t) = s_2(t) \quad si \quad s_0(t) = 1.$$

La configuració d'un LFSR s'estableix mitjançant el seu grau (o nombre de cel·les del registre) i el seu polinomi de connexions (en altres paraules, el conjunt de les seves portes lògiques). El problema plantejat aquí, doncs, consisteix en determinar, a partir de la contrasenya donada per l'usuari:

a) Per una part, **els graus i els polinomis de connexions dels 3 LFSR**.

b) Per una altra part, **els estats inicials dels 3 LFSR**.

La seqüència pseudoaleatòria de sortida (fins al nombre de bits necessaris) serà la clau criptogràfica que es farà entrar en el xifrador de bloc.

De les incomptables solucions que pot tenir el problema plantejat, n'escolliré la que exposo a continuació. Correspon a una de les pràctiques que vaig realitzar en cursar l'assignatura de Criptografia durant el segon semestre del 2004 a la UOC.

Grau dels LFSR

- Tots tres LFSR tindràn el mateix grau. Sigui g = grau LFSR.

Polinomis de connexions

De cada un dels caràcters de la contrasenya se n'extreuen els 4 bits de menys pes (els de més a la dreta) de la seva codificació Unicode, i es concatenen.

Després, la cadena de bits així obtinguda es divideix en 3 blocs de g-1 bits i en 3 blocs més de g bits.

Llavors,

- Polinomi de connexions del LFSR1

  Es definirà amb el **primer bloc de g-1 bits**. El coeficient de més pes del polinomi es fixa a 1, i el terme independent també. Els valors de la resta de coeficients C_i s'assignen per ordre creixent: C_1 igual al primer bit de la cadena d'entrada, C_2 al segon, etc.

  Per exemple, si g = 4 i la cadena de bits obtinguda a partir de la contrasenya és 1001111010010101111, el polinomi de connexions de LFSR1 serà: $C(x) = 1 + 1 \cdot x + 0 \cdot x^2 + 0 \cdot x^3 + 1 \cdot x^4$

- Polinomi de connexions del LFSR2

  Es definirà amb el **segon bloc de g-1 bits**, de la mateixa manera que abans.

- Polinomi de connexions del LFSR0

  Amb el **tercer bloc de g-1**, com en els dos LFSR anteriors.

Estats inicials dels LFSR

- Estat inicial del LFSR1

  Es definirà amb el quart bloc de g bits. El primer bit es ficarà a la cel·la S_1, el segon a la S_2, etc.

- Estat inicial del LFSR2

  Amb el cinquè bloc de g bits, com abans.

- Estat inicial del LFSR0

  Amb el sisè bloc de g bits.

La següent figura il·lustra com es distribueixen els bits de la cadena obtinguda a partir de la contrasenya per definir els polinomis C(n) de connexió i els estats inicials S(n) dels LFSR.

g-1 bits	g-1 bits	g-1 bits	g bits	g bits	g bits	...
↓	↓	↓	↓	↓	↓	
C(n)LFSR1	C(n)LFSR2	C(n)LFSR0	S(n)LFSR1	S(n)LFSR2	S(n)LFSR0	

La implementació d'aquest mètode de construcció del generador Geffe no admetrà les contrasenyes que portin a algun dels 3 LFSR cap a un estat absorbent.

Finalment, dir que tots 3 LFSR seran de grau 8 (és a dir, g = 8), i que les claus criptogràfiques generades hauran de tenir una longitud depenent de l'algorisme de xifratge escollit.

3.5 Diagrames de classes

Les classes que formen part del programa les he agrupades en 3 paquets:

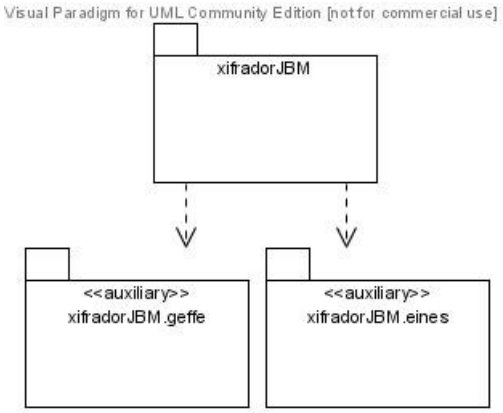


Figura 37: Diagrama dels paquets de l'aplicació

El paquet ***xifradorJBM*** conté les classes principals de l'aplicació. Per una altra part, el paquet ***xifradorJBM.geffe*** conté les classes corresponents al generador Geffe per a l'obtenció de les claus criptogràfiques a partir de la contrasenya donada per l'usuari. Per últim, el paquet ***xifradorJBM.eines*** conté algunes classes auxiliars que es fan servir durant el processos de xifratge i desxifratge.

Els diagrames de classes de cada un d'aquest paquets, per tal de codificar en una etapa posterior un programa l'execució del qual es comporti d'acord amb la solució proposada en els apartats anteriors, els he dissenyat de la següent manera:

3.5.1 Paquet xifradorJBM

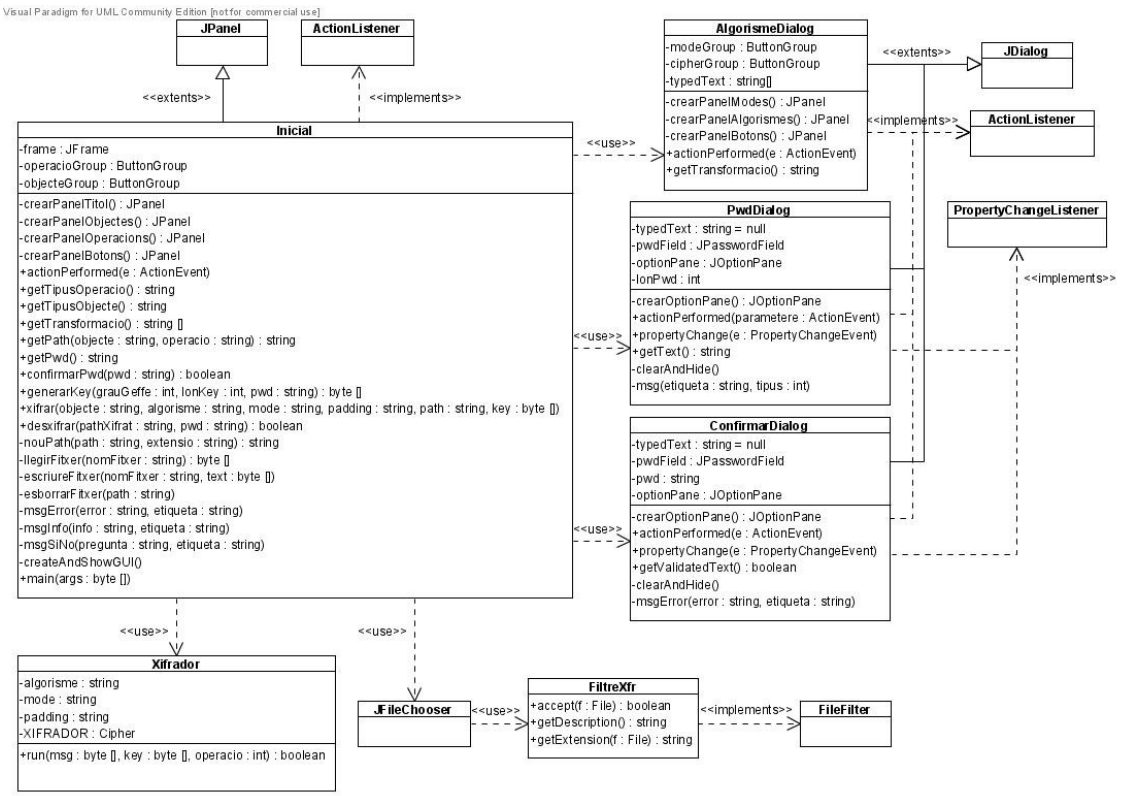
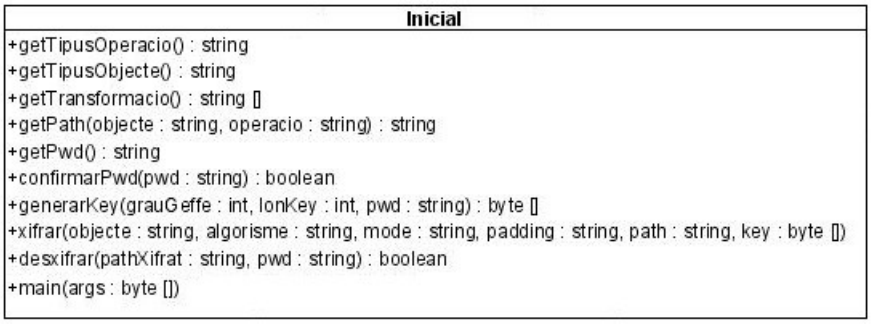


Figura 38: Diagrama de classes del paquet xifradorJBM

En els següents subapartats faré una breu descripció de cada una de les classes i mostraré les seves interfícies (mètodes públics).

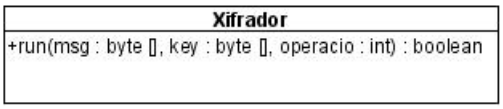
3.5.1.1 Classe Inicial

Es correspon amb la pantalla inicial des de la qual l'usuari podrà xifrar o desxifrar arxius o carpetes. Hereta de la classe javax.swing.JPanel i conté els mètodes mitjançant els quals el programa obtindrà de l'usuari els paràmetres necessaris per al xifratge: el tipus d'operació, el tipus d'objecte, l'algorisme de xifratge, etc.



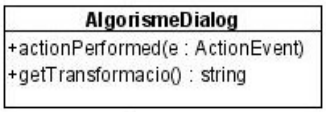
3.5.1.2 Classe Xifrador

Implementa el xifrador de bloc, el tipus del qual estarà determinat per l'atribut *transformacio* (DES, 3DES, IDEA, Blowfish o AES). Per instanciar-ne un caldrà especificar valors per als atributs *transformacio* i *tipusOperacio* (xifratge o desxifratge).



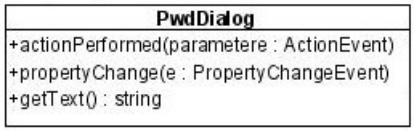
3.5.1.3 Classe AlgorismeDialog

Es correspon amb el quadre de diàleg mitjançant el qual l'usuari podrà seleccionar l'algorisme de xifratge i el mode d'operació. Hereta de la classe javax.swing.JDialog.



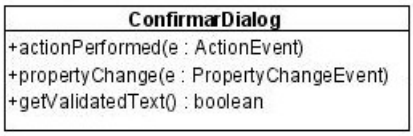
3.5.1.4 Classe PwdDialog

Es correspon amb el quadre de diàleg a través del qual l'usuari podrà introduir la contrasenya per xifrar o desxifrar



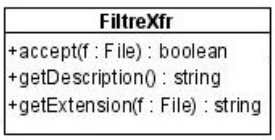
3.5.1.5 Classe ConfirmarDialog

Es correspon amb el quadre de diàleg a través del qual l'usuari podrà repetir la contrasenya introduïda en un altre quadre de diàleg anterior per tal de confirmar-la abans de xifrar.



3.5.1.6 Classe FiltreXfr

Aquesta classe implementa la classe javax.swing.filechooser.FileFilter. Si l'operació seleccionada per l'usuari és desxifrar un arxiu, l'explorador permetrà filtrar només els arxius amb l'extensió que el xifrador els assigna per defecte (per exemple, .xfr). Això facilitarà la cerca dels fitxer xifrats.



3.5.2 Paquet xifradorJBM.Geffe

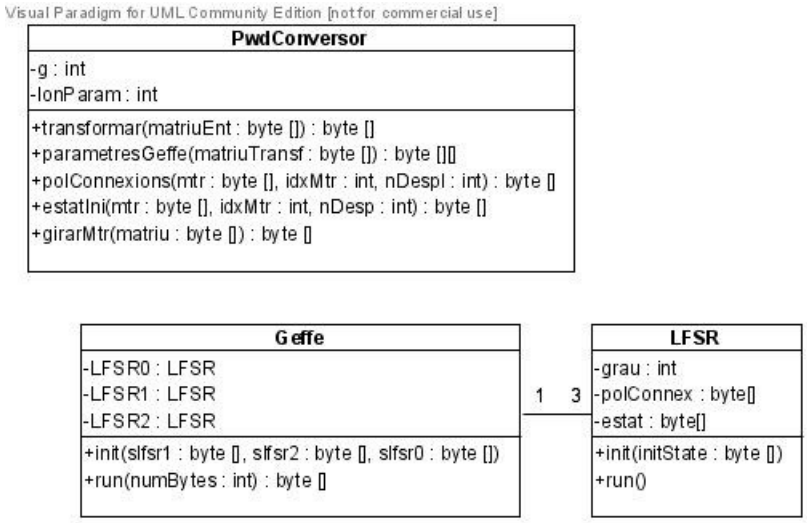
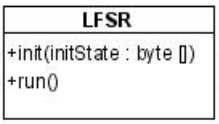


Figura 39: Diagrama de classes del paquet xifradorJBM.Geffe

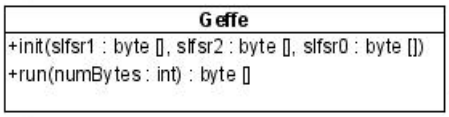
3.5.2.1 Classe LFSR

Aquesta classe implementa un generador pseudoaleatori LFSR. Un LFSR es caracteritza pel seu grau i el seu polinomi de connexions. Per tant, en instanciar-ne un, aquests són els paràmetres que caldrà especificar.



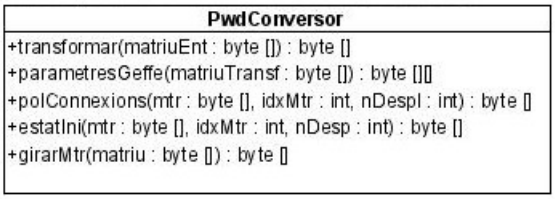
3.5.2.2 Classe Geffe

Implementa el generador Geffe per a la obtenció de les claus criptogràfiques. Tal com ja he comentat en un altre apartat, es construeix amb 3 LFSR.



3.5.2.3 Classe PwdConversor

Es tracta d'una classe auxiliar mitjançant la qual s'obtindran, a partir de la contrasenya que hagi donat l'usuari, els polinomis de connexions i els estats inicials dels 3 LFSR del generador Geffe. A l'apartat 5 s'ha descrit en què consisteix aquest procés.



3.5.3 Paquet xifradorJBM.eines

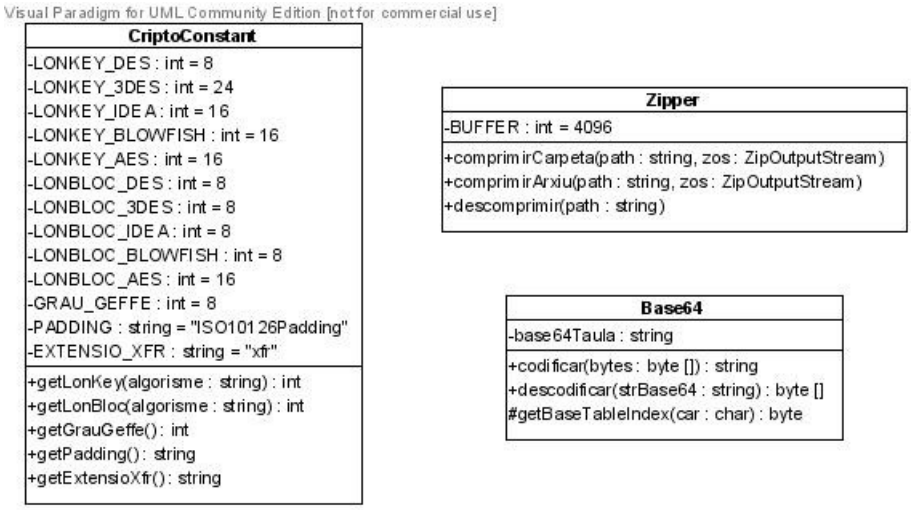
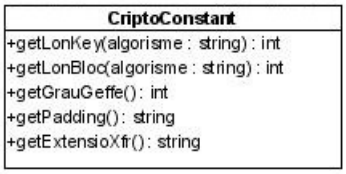


Figura 40: Diagrama de classes del paquet xifradorJBM.eines

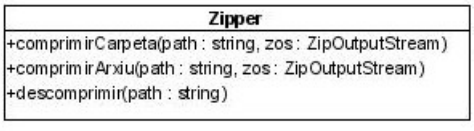
3.5.3.1 CriptoConstant

Conté constants utilitzades en unes altres parts de l'aplicació, permetent obtenir-les.



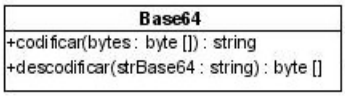
3.5.3.2 Zipper

Compressor que treballa en format zip.



3.5.3.3 Base64

Permet codificar en Base64 una representació ASCII, i inversament, descodificar una cadena Base64 a ASCII.



4 Conclusions

Durant la implementació de l'aplicació em vaig veure obligat a introduir alguns canvis en el disseny respecte del proposat en el document de disseny lliurat unes setmanes abans. El més obvi consisteix en haver classificat les classes en tres paquets en lloc d'incloure-les totes en un únic paquet. Un d'aquests paquets conté les classes corresponents al generador Geffe usat per a l'obtenció de la clau criptogràfica a partir de la contrasenya. En una altra hi ha classes que es poden considerar eines utilitzades en el procés de xifratge i desxifratge: un compressor (o zipper), un codificador de Base64, i una classe que conté constants criptogràfiques referenciades en unes altres parts de l'aplicació. El tercer paquet conté les classes que constitueixen el nucli del programa.

Pel que fa a comprimir els arxius abans de xifrar-los, en un primer moment, i segons m'ho va suggerir el meu consultor, només tenia previst fer-ho sobre les carpetes, però tenint en compte el benefici que suposa la reducció d'espai en comprimir els fitxers, tant si només es vol emmagatzemar-los com si es tracta d'intercanviar-los a través de la xarxa, vaig prendre la decisió de comprimir no només les carpetes sino també els arxius.

També vaig decidir desar els fitxers xifrats en format XML amb la finalitat de facilitar-li a l'usuari el desxifratge. Em va sorgir llavors un problema: d'acord

amb les especificacions http://www.w3.org/TR/1998/REC-xml-199880210 del consorci W3C del 10 de febrer de 1998 hi ha cert caràcters no admesos en la sintaxi XML, la qual cosa ocasionava una excepció en "parsejar" el document durant el procés de desxifratge. La primera solució que hi vaig donar va ser codificar la xifra en hexadecimal abans de crear el document XML. Però això implicava duplicar el seu tamany. Després de consultar alguns fórums, vaig optar per codificar la xifra en Base64.

Un altre problema amb el que m'hi he trobat és que per culpa d'una lectura precipitada de la documentació de Java vaig creure que el proveïdor SunJCE proporcionava també, a més del DES, 3DES, Blowfish i AES, l'algorisme IDEA. Durant la implementació vaig poder adonar-me del malentés. He deixat pendent d'importar aquest algorisme d'algun altre proveïdor, o intentar implementar-lo pel meu compte.

A última hora vaig trobar-me que l'aplicació no era capaç de xifrar arxius superiors a 3 Mbytes. Fins llavors, les proves les havia anat fent amb fitxers més petits. La informació era transmesa d'una fase a la següent mitjançant arrays, per la qual cosa, si un arxiu superava aquest tamany, el programa no disposava de suficient memòria RAM per poder-se executar correctament. Per solucionar aquest problema he hagut de substituir la manera de transferir la informació d'una fase a la següent amb l'ús de streamers, creant fitxer temporals en el disc dur.

Vull afegir-hi que, per afers personals, vaig lliurar el document de disseny amb uns dies de retard. Abans, però, l'hi vaig demanar al meu consultor. Ell va comprende la meva situació i em va respondre afirmativament a la petició que li feia, animant-me a continuar el treball. Aprofito per a donar-li les gràcies.

Finalment, pel que fa a la valoració personal que faig d'aquest TFC, només dir que el funcionament de l'aplicació resultant de la fase de disseny satisfà les espectatives plantejades inicialment en el pla de treball elaborat al començament de tot.

5 Glossari

Bloc: seqüència de bits d'una certa longitud que representa la unitat de xifratge d'un criptosistema de bloc.

Clau criptogràfica: cadena de caràcters emprada per xifrar un text en clar.

Confusió: tècnica consistent en incrementar el grau de complexitat de la relació existent entre la clau criptogràfica i el text xifrat.

Criptosistema (o sistema criptogràfic): dispositiu implementat bé en software, bé en hardware la finalitat del qual consisteix en el xifratge d'arxius (en format de text, imatge, so, etc.).

Difusió: tècnica consistent en dissipar les propietats estadístiques del text en clar.

Esquema Feistel: esquema SPN que al mateix temps serveix tant per xifrar com per a desxifrar.

Esquema SPN (o xarxa SPN): estructura d'un algorisme de xifratge que utilitza taules de substitucions i taules de permutacions (Substitution-Permutation Network).

Generador Geffe: NLFSR construït amb 3 LFSR, dos dels quals generen dues seqüències i el tercer determina el valor de sortida de la funció.

LFSR: dispositiu format per un cert nombre de cel·les de memòria i de portes lògiques, el qual constitueix el xifrador de flux més elemental de tots (Linear Feedback Shift Register o registre de desplaçament realimentat linealment).

Mode d'operació: manera com són introduïts en un xifrador de bloc la seqüència de blocs en la que ha estat dividit el text en clar (ECB, CBC, CFB, OFB i CTR).

NLFSR: xifrador de flux format per varis LFSR la sortida dels quals serveixen d'entrada per a un altre dispositiu que té el comportament d'una funció no lineal (Non Linear Feedback Shift Register).

Substitució: tècnica consistent en substituir per uns altres, les lletres, dígits o símbols d'un text en clar.

Taula de permutacions (o P-box): taula utilitzada per un algorisme de xifratge per a generar difusió.

Taula de substitucions (o S-box): taula utilitzada per un algorisme de xifratge per a generar confusió.

Text en clar: Text que es vol xifrar.

Transposició: tècnica consistent en reordenar dins un mateix text les seves lletres, dígits o símbols.

Xifra: Text resultant de xifrar un text en clar.

Xifrador de bloc: xifrador de clau simètrica que xifra un bloc rere bloc.

Xifrador de clau asimètrica (o de clau pública): xifrador que fa servir diferents claus criptogràfiques per al xifratge i per al desxifratge.

Xifrador de clau simètrica (o de clau compartida): xifrador que fa servir la mateixa clau criptogràfica tant per al xifratge com per al desxifratge (Per exemple, DES, 3DES, IDEA, Blowfish, AES).

Xifrador de flux: xifrador de clau simètrica en el que el text en clar va xifrant-se com si es tractés d'un flux de bits.

6 Bibliografia i recursos

Llibres.-

Campderrich Falgueras, Benet (2004). *Enginyeria del programari.* UOC

Domingo Ferrer, J. [et al.] (2004) *Criptografia*. UOC

Minguillón i Alfonso, Julià (2003). *Fonaments de programació II*. UOC.

Scheier, Bruce (1996) *Applied Cryptography*. John Wiley & Sons, Inc.

Stinson, Douglas R. (2006) *Cryptography. Theory and Practice*. Chapman & Hall/CRC.

Enllaços.-

http://en.wikipedia.org/wiki/Symmetric-key_algorithm

http://en.wikipedia.org/wiki/Block_cipher

http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

http://en.wikipedia.org/wiki/Data_Encryption_Standard

http://en.wikipedia.org/wiki/Triple_DES

http://en.wikipedia.org/wiki/Blowfish_%28cipher%29

http://en.wikipedia.org/wiki/International_Data_Encryption_Algorithm

http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

http://www.nist.gov/

http://csrc.nist.gov/CryptoToolkit/aes/rijndael/

http://www.iaik.tu-graz.ac.at/research/krypto/AES/

http://fp.gladman.plus.com/AES/index.htm

http://theory.lcs.mit.edu/~rivest/crypto-security.html

http://www.kriptopolis.com

http://www.mis-algoritmos.com/ejemplos/source-164.html

Recursos

- Per a l'elaboració dels diagrames de casos d'ús, d'activitats i de classes he utilitzat l'aplicació *Visual Paradigm for UML Community Edition (version 6.0)*. Es pot descarregar la versió de prova de l'enllaç:

  http://www.visual-paradigm.com

- La interfície gràfica d'usuari l'he dissenyada mitjançant els paquets *java.awt* i *javax.swing*. He utilitzat l'entorn de programació *Eclipse SDK (version 3.2.2)*. Es pot descarregar de l'enllaç:

  http://www.eclipse.org

- L'assistent per a la instal·lació del programa l'he creat amb la versió Freeware de l'**Advanced Installer 4.9.2**, la qual es pot descarregar de

  http://www.advancedinstaller.com

Il·lustracions

Les figures de la 1 a la 24 les he extretes de l'enciclopèdia:

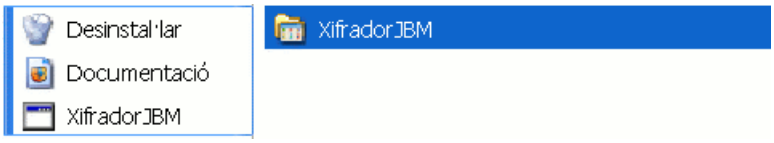
http://en.wikipedia.org

7 Annex: Manual d'instal·lació de l'aplicació

7.1 Introducció

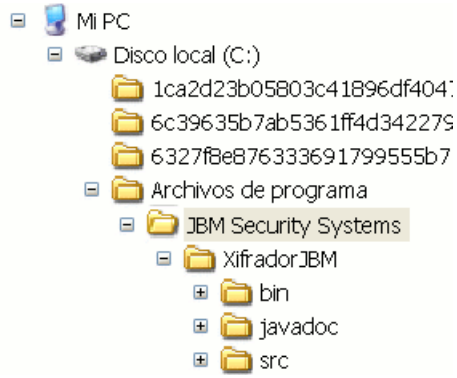
Amb la versió de prova del programa **Advanced Installer 4.9.2** he generat l'assistent que permetrà integrar l'aplicació *XifradorJBM* d'aquest projecte en l'entorn Windows XP sobre el que s'hagi d'executar. Finalitzada la instal·lació:

- Des del **menú Inici** l'usuari podrà arrencar el programa, a més de poder accedir a la documentació Javadoc. També se li oferirà la possibilitat de desinstal·lar-lo des d'aquest menú.



- A l'**escriptori** s'hi haurà creat un accés directe cap a l'executable de l'aplicació.

- Per defecte, l'executable i el Javadoc s'instal·laran en **Archivos de programa\ JBM Security Systems\ XifradorJBM**. En aquesta mateixa carpeta s'instal·laran també, tot i no ser necessari, les classes (en \bin) i el codi font (en \src).



- L'aplicació també s'obrirà en fer doble clic sobre els arxius amb extensió **xfr**.

7.2. Requisits

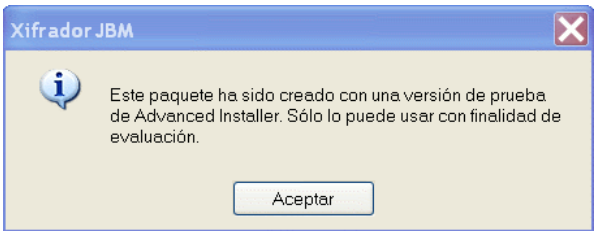
Aquest assistent permetrà la instal·lació del xifrador en entorns Windows 95, Windows Millennium, Windows NT, Windows 2000, Windows XP i Windows Vista.

Per una altra part, caldrà tenir instal·lat el JRE 1.5 (o una versió superior).

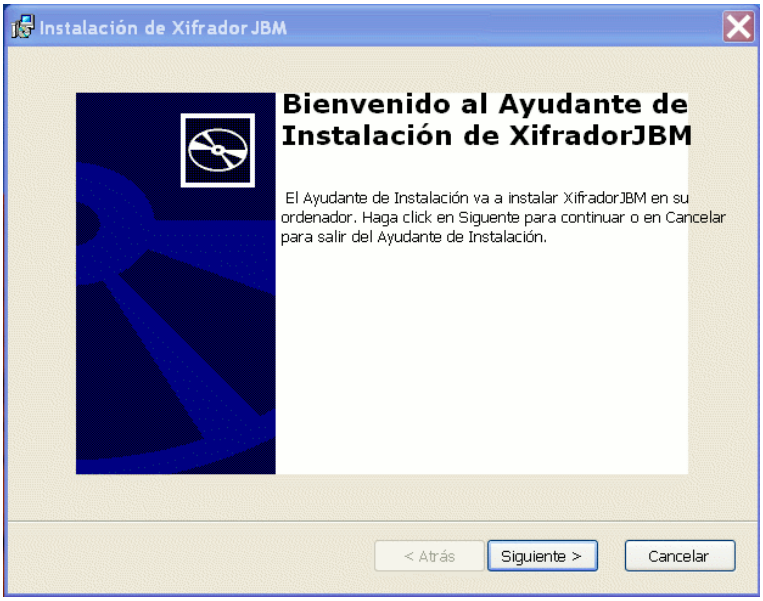
7.3. Descripció

El nom que li he donat a l'assistent és XifradorJBM.msi. La instal·lació del programa XifradorJBM s'iniciarà fent doble clic sobre la seva icona:

Un missatge li adverteix a l'usuari que aquest assistent ha estat creat amb una versió de prova i que, per tant, només podrà ser usat amb finalitats d'avaluació.

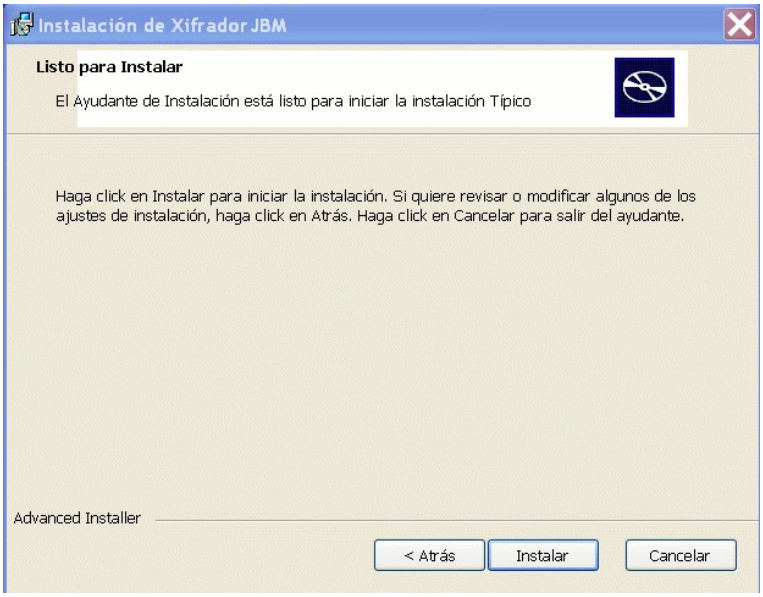


Tot seguit apareix un missatge de benvinguda:

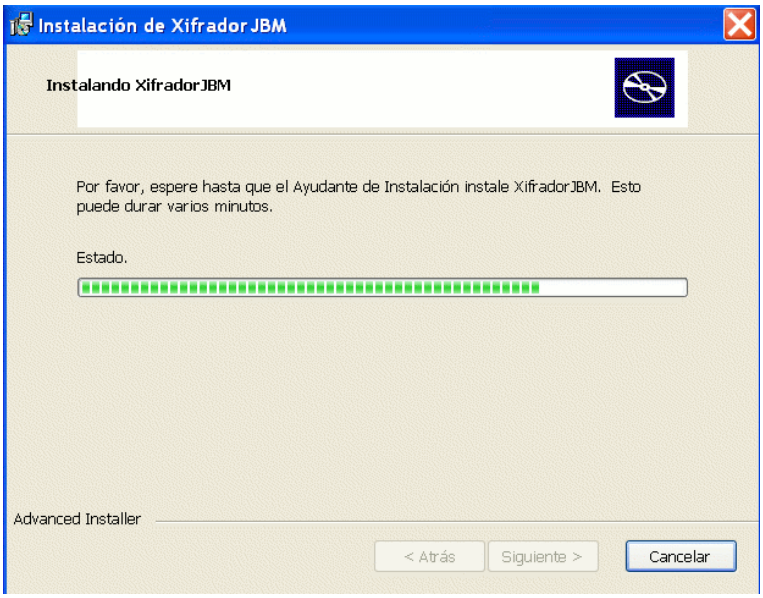


En el següent quadre de diàleg l'usuari podrà modificar la carpeta on s'instalaran l'executable, el Javadoc, les classes i el codi font.

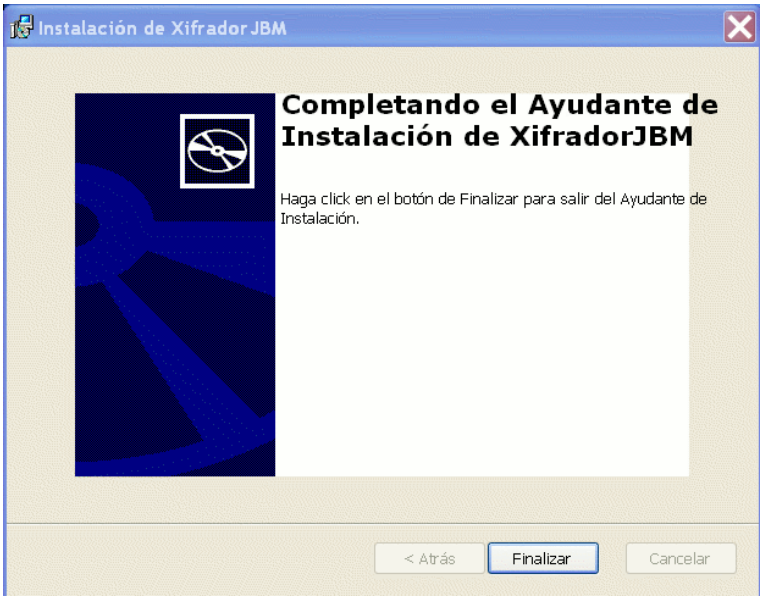
L'assistent ja es troba preparat per a iniciar la instal·lació. L'usuari haurà de confirmar si vol continuar:



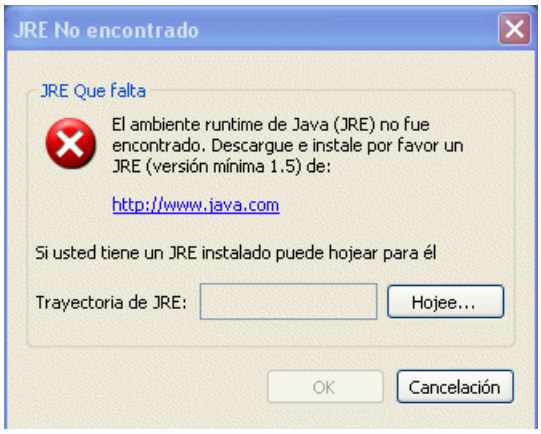
Una altra finestra de diàleg li indica a l'usuari l'evolució de la instal·lació, permetent-li cancel·lar-la si així ho desitja.



Quan finalment s'ha completat la instal·lació, l'assistent li ho indica a l'usuari.



L'execució del xifrador exigeix que hi hagi instal·lat el JRE 1.5. Si aquest no fos el cas, en intentar executar-lo la primera vegada s'obrira aquesta finestra amb un enllaç des d'on l'usuari podrà descarregar-se'l:


JRE No encontrado
JRE Que falta
El ambiente runtime de Java (JRE) no fue encontrado. Descargue e instale por favor un JRE (versión mínima 1.5) de:
http://www.java.com
Si usted tiene un JRE instalado puede hojear para él
Trayectoria de JRE:
Hojee...
OK
Cancelación