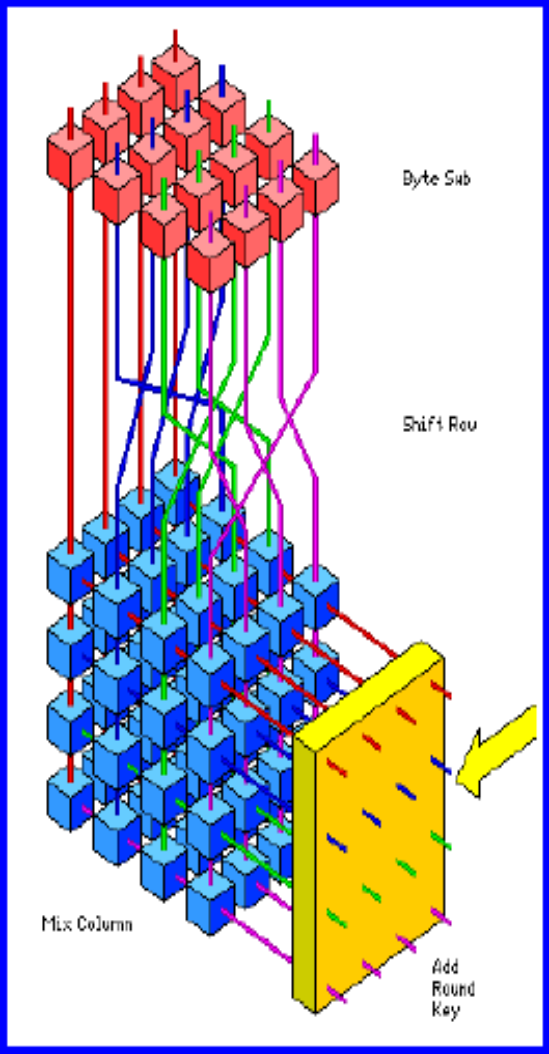
Securitatea transmiterii informației pe INTERNET

Îndrumător de lucrări de laborator



Kovaci Maria

Alexandru Isar

Cuprins

Lucrarea nr.1.

Metoda de criptare RSA

1. Introducere

Este o metodă de criptare cu chei publice. Ideea de bază constă în faptul că procedura de criptare este facută publică de către fiecare utilizator şi poate fi folosită de toţi ceilalţi utilizatori pentru cifrarea mesajelor ce le sunt adresate. În schimb procedura de decriptare, diferită de cea de criptare, este ţinută secretă. Cele două proceduri se aleg astfel încât cunoaşterea uneia (cea făcută publică) să nu permită obţinerea celei de a doua (cea de descifrare).

2. Aritmetica în câmpul claselor de resturi modulo număr prim

Criptosistemul RSA foloseşte exponenţierea în câmpul claselor de resturi modulo numar prim pentru a cripta şi decripta mesaje text convertite într-o formă numerică.

În cele ce urmează se prezintă ideile principale privind aritmetica în acest câmp, utilizate şi de către metoda RSA.

Aritmetica modulară este o aritmetică cu numere întregi şi pozitive (sau zero). Aritmetica modulară este similară cu cea obişnuită cu o singură diferenţă: aritmetica modulară lucrează cu o mulţime restrânsă de numere, mulţime definită de către un singur număr - modulul.

De exemplu, să considerăm aritmetica modulo 5 (pentru simplitate). Mulţimea de numere utilizată este {0,1,2,3,4}. În această clasă, considerând numerele în ordine crescătoare după cifra 4 urmează cifra 0 şi apoi 1 ... numerele repetându-se ciclic, cu perioada cinci. Orice număr în această clasă se transformă într-unul din cele cinci numere posibile. De fapt el este egal cu restul împărţirii lui la 5. În aritmetica câmpului modulo 5 numerele se transformă astfel:

0 -> 0
1 -> 1
2 -> 2
3 -> 3
4 -> 4
5 -> 0
6 -> 1
7 -> 2
8 -> 3
...

2.1. Înmulţirea modulo număr prim

Înmulţirea a două numere modulo 5 este destul de simplă. În primul rând ne bazăm doar pe mulţimea de numere {0,1,2,3,4}.

Numerele se înmulţesc în mod obişnuit dar se reţine ca rezultat restul împărţirii la 5 a produsului obţinut. Astfel 3*3 mod 5 = 4, deorece 3*3 = 9 iar 9 mod 5 este 4.

Putem stabili tabela înmulţirii modulo 5, astfel:

| 0 1 2 3 4
---
0 | 0 0 0 0 0
1 | 0 1 2 3 4
2 | 0 2 4 1 3
3 | 0 3 1 4 2
4 | 0 4 3 2 1

Generalizarea pentru orice modul este foarte simplă. Orice număr reprezentat modulo n este egal cu restul împărțirii lui cu numărul n.

2.2. Proprietăți ale multiplicării modulo număr prim

1. Identitatea

Orice număr din această clasă înmulțit cu 1 dă tot numărul respectiv.Astfel,

1*1 mod 5 = 1
2*1 mod 5 = 2
3*1 mod 5 = 3 .. etc

2. Inversa

Un număr N are o inversă M, dacă N înmulțit cu M dă elementul identitate.

Astfel pentru modulo 5:

3*2 mod 5 = 1,

deci numărul 2 este inversul numărului 3 pentru înmulțirea modulo 5.

3. Numere relativ prime şi inversa multiplicativă

În exemplul de mai sus toate numerele, exceptând numărul 0, au inversă în raport cu înmulțirea (multiplicativă). Acest lucru însă nu este adevărat pentru orice valoare de modul. Astfel, s-a arătat că un număr M oarecare are inversă multiplicativă în raport cu înmulțirea modulo N, doar dacă cele două numere sunt relativ prime. În cazul înmulțirii modulo 5, acest deziderat se verifică, deorece N=5 este un număr prim.

4. Funcția indicatorul lui Euler

Funcția indicatorul lui Euler, în cazul unui modul N, J(N), indică numărul numerelor relativ prime cu N din mulțimea {1 ... N-1}. Astfel, ea indică câte numere din această mulțime posedă inversă multiplicativă, în raport cu numărul N.

Funcția indicatorul lui Euler, în cazul unui modul N, J(N), poate fi calculată factorizând numărul N.

De exemplu:
Dacă, N=P1*P2* ...*PM (P1 ...PM - numere prime)
atunci: J(N)=(P1-1)*(P2-1)* ...*(PM-1)

Dacă N este un număr prim, rezultă J(N)=N-1

2.3. Exponențierea modulară şi funcţia indicatorul lui Euler

Vom nota cu simbolul "^" operatorul de exponențiere. Cum se calculeaza o anumită putere modulo un anumit număr ? Să considerăm din nou N=5. Calculăm 3^2 mod 5: 3^2 = 9 iar 9 mod 5 este 4 .

Exponeţierea modulo un anumit număr poate fi calculată şi iterativ, astfel:

(3^3 mod 5)^2 mod 5 = (3^3)^2 mod 5

dar:

(3^3 mod 5)^2 mod 5 = 2^2 mod 5=4

În aritmetica obişnuită exponențierea iterată se reduce la o singură exponențiere. Astfel: (3^3)^2 = 3^6 deorece 3*2 = 6.

În aritmetica modulo 5 (unde nu există numărul 6): (3^3)^2 mod 5 devine 3^(3*2 mod J(5)) mod 5, care, deorece J(5) = 4, este 3^2 mod 5 = 4.

Sistemul de criptare RSA este de tip exponențial. Funcția de criptare RSA este:

c=w^e mod n

unde:

n - modulul,
e - exponentul public,
w - textul clar,
c - textul criptat,

Modulul n este obținut prin produsul a două numere prime mari:

n=p*q

astfel încât funcția indicatorul lui Euler:

J(n)=(p-1)*(q-1)

devine mult mai greu de determinat.

Perechea (e,n) este făcută publică (cheia publică). Funcția de decriptare RSA este:

w=c^d mod n

unde:

d - exponentul privat, ce verifică: e*d mod J(n) = 1, deci d reprezintă inversul multiplicativ al lui e în raport cu funcția indicatorul lui Euler, J(n).

Acest lucru înseamnă, deasemenea, că numărul e este inversul lui d față de operația de exponențiere modulo n. Astfel:

(w^e mod n)^d mod n = w^(e*d mod J(n)) mod n = w

Găsirea lui d (valoarea lui e fiind publică) necesită cunoaşterea lui J(n) pentru n fixat. Rezistența la atacuri a metodei de criptare constă tocmai în dificultatea factorizării numărului n (pentru o valoare suficient de mare).

3. Proiectarea unui sistem de criptare RSA

Dimensionarea unui astfel de sistem porneşte în mod curent de la alegerea a două numere prime mari: p şi q.

Se calculează:

$$n=p*q$$

şi:

$$J(n)=(p-1)*(q-1)$$

Se alege un număr e, relativ prim cu J(n). Se recomandă alegerea lui e din intervalul [max(p,q)+1, J(n)]. Se caută numărul d, inversul față de exponențierea modulo n al numărului e, număr ce verifică:

$$w^{(e*d)} \bmod n = w$$

pentru orice w din mulțimea {1 ... n-1}, problemă ce se reduce la a căuta inversul față de multiplicarea modulo J(n), deorece:

$$e*d \bmod J(n) = 1$$

Având aceste numere, se face publică perechea (e,n) - cheia publică şi se păstrează secret cuplul (d,n) - cheia privată.

Criptarea presupune calculul valorii:

$$c = w^e \bmod n$$

în timp ce decriptarea se face prin calculul valorii:

$$w = c^d \bmod n$$

4. Un exemplu de utilizare a sistemului de criptare RSA

Alegem:

$$p = 7$$
$$q = 11$$

rezultă:

$$n = p*q = 77 \quad \text{- modulul}$$
$$J(n) = (p-1)*(q-1) = 60 \quad \text{- funcția indicatorul lui Euler}$$

Alegem:

$$e = 37 \quad \text{- un număr relativ prim cu J(n)}$$

rezultă:

$$d = 13 \quad \text{- un număr ce verifică relația}$$

$$e*d \bmod J(n) = 1$$

Considerăm corespondența dintre alfabetul latin şi cifre:

A B C D E F G H I J K L M N O P R S T U V W X Y Z _
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
unde caracterul '_' semnifică 'spațiu'.

Deorece n < 2626 criptarea se va face caracter cu caracter. Astfel alegând, de exemplu, textul clar, 'H', lui îi va corespunde cifra, w=8.
Textul criptat se determină conform:

$$c = w^e \bmod n$$

În exemplul nostru:

$$c = 8^{37} \bmod 77 = (8^{32} * 8^4 * 8) \bmod 77 =$$
$$= ((8^{32} \bmod 77)*(8^4 \bmod 77)*(8 \bmod 77)) \bmod 77 =$$
$$= 64*15*8 \bmod 77 = 57$$

$c = 57$

unde exponențialele modulo 77 succesive ale lui 8 se calculează în mod iterativ, calculând pe rând:

$$8 \bmod 77 = 8$$

$$8^2 \bmod 77 = 64$$
$$8^4 \bmod 77 = 8^2*8^2 \bmod 77 =$$
$$= ((8^2 \bmod 77)*(8^2 \bmod 77)) \bmod 77 = 15$$

$$8^8 \bmod 77 = 8^4*8^4 \bmod 77 =$$
$$= ((8^4 \bmod 77)*(8^4 \bmod 77)) \bmod 77 =$$
$$= 15*15 \bmod 77 = 71$$

$$8^{16} \bmod 77 = 8^8*8^8 \bmod 77 =$$
$$= ((8^8 \bmod 77)*(8^8 \bmod 77)) \bmod 77 =$$
$$= 71*71 \bmod 77 = 36$$

$$8^{32} \bmod 77 = 8^{16}*8^{16} \bmod 77 =$$
$$= ((8^{61} \bmod 77)*(8^{16} \bmod 77)) \bmod 77 =$$
$$= 36*36 \bmod 77 = 64$$

Decriptarea presupune calculul:

$$w = c^d \bmod n$$

În cazul nostru:

$$w = 57^{13} \bmod 77 = (57^8 * 57^4 * 57) \bmod 77 =$$
$$= ((57^8 \bmod 77)*(57^4 \bmod 77)*(57 \bmod 77)) \bmod 77 =$$
$$= 36*71*57 \bmod 77 = 8$$

$w = 8$

unde exponențialele modulo 77 succesive ale lui 57 se calculează deasemenea în mod iterativ, calculând pe rând:

$$57 \bmod 77 = 57$$

$$57^2 \bmod 77 = 15$$
$$57^4 \bmod 77 = (57^2 * 57^2) \bmod 77 =$$
$$= ((57^2 \bmod 77) * (57^2 \bmod 77)) \bmod 77 =$$
$$= 15 * 15 \bmod 77 = 71$$

$$57^8 \bmod 77 = (57^4 * 57^4) \bmod 77 =$$
$$= ((57^4 \bmod 77) * (57^4 \bmod 77)) \bmod 77 =$$
$$= 71 * 71 \bmod 77 = 36$$

5. Prezentarea programului folosit

Metoda de criptare RSA este simulată cu ajutorul unui program scris în limbajul C. Lansarea acestui program se efectuează din Windows Commander făcând *click* pe Lab_crypt.exe.
Pe ecran apare un panou virtual de sistem de criptare RSA, cum este cel din figura 1.

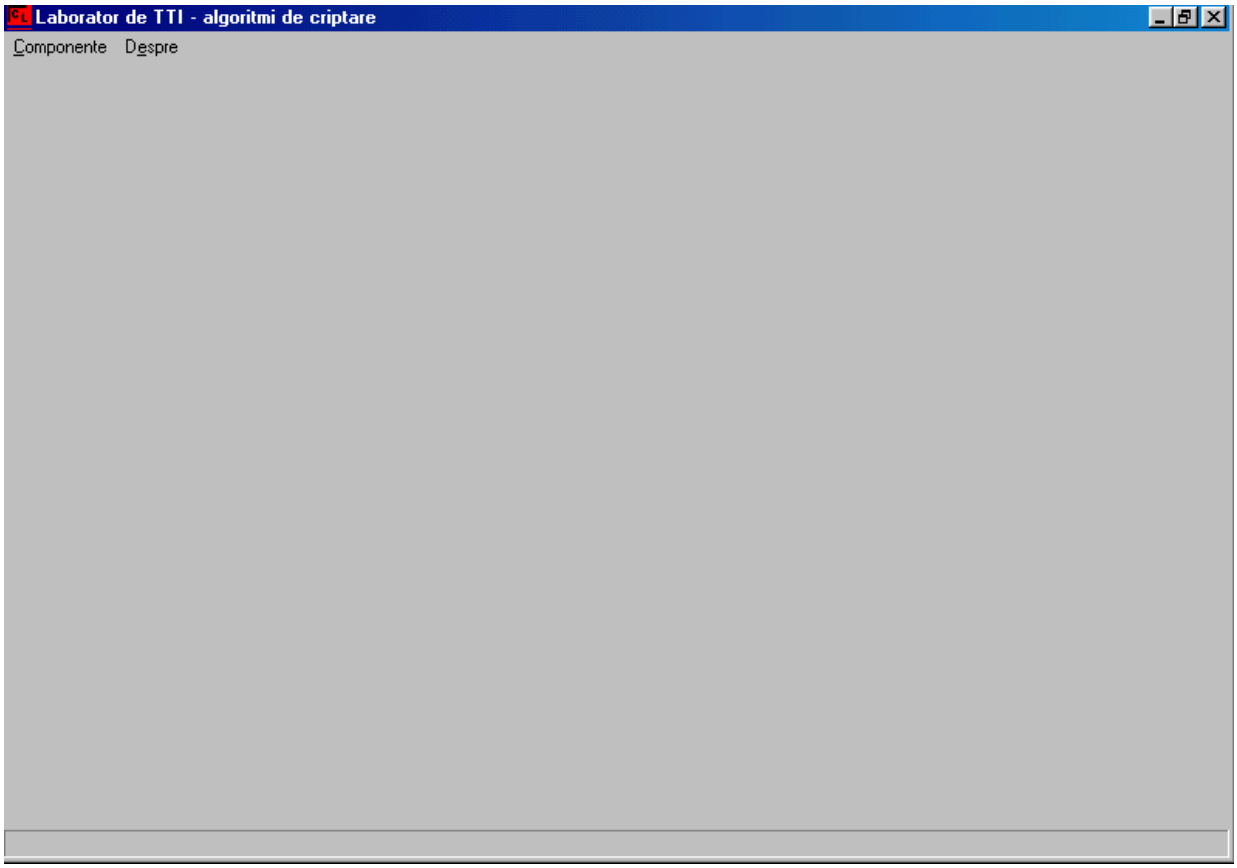


Figura 1. Panoul frontal al sistemului virtual de criptare RSA.

Principalele funcții ale acestui sistem sunt:

Componente - se prezintă algoritmii implementați.
Făcând *click* pe butonul Componente de pe panoul frontal, se deschide fereastra prezentată în figura 2.

Figura 2. Subfuncțiile corespunzătoare funcției Componente.

Făcând *click* pe butonul Algoritmul RSA, se deschide fereastra prezentată în figura următoare.

Figura 3. Subfuncțiile corespunzătoare funcției Algoritmul RSA.

Făcând *click* pe butonul Descriere poate fi citită o prezentare a algoritmului RSA. Făcând *click* pe butonul Exemplu poate fi urmărită realizarea unei criptări şi a decriptării corespunzătoare.
Făcând *click* pe butonul Exercițiu pot fi realizate criptări şi decriptări cu algoritmul RSA.
În continuare se prezintă un exemplu pentru înțelegerea modului de utilizare a acestei funcții.
Folosind ferestrele de sub literele p şi q se aleg pentru acestea valorile 7 şi 11. După apăsarea butonului Generează rezultă valorile 7 pentru E şi 43 pentru D, cu ajutorul cărora se construiesc cheia publică (7,77) şi cheia privată (43,77). După apăsarea butonului > poate fi realizată criptarea unei litere cu algoritmul RSA. Se alege, de exemplu, folosind fereastra de sub mesajul Alegeți o literă de criptat, litera H. Se specifică cheia publică de criptare (7,77). Se apasă butonul Criptează şi se obține mesajul criptat 57. În vederea decriptării acestuia se apasă butonul >. În fereastra corespunzătoare mesajului criptat se scrie 57. În fereastra Cheia privată se scrie (43,77). Se apasă butonul Decriptează. Se obține mesajul original, decriptat cu cheia privată, H.

6. Desfăşurarea lucrării

6.1. Se verifică toate funcțiile programului evidențiate în paragraful anterior. În acest mod se învață să se lucreze cu programul
6.2. Se efectuează verificarea funcționării programului efectuând o criptare şi o decriptare. Rezultatele vor fi salvate într-un fişier word, din directorul user, in al carui nume vor aparea numele studentului şi L1.

6.3. Pentru acelaşi mesaj se va simula folosirea a două sisteme de criptare RSA, prin alegerea, a două seturi diferite de parametri p, q şi e. Pentru fiecare dintre aceste experimente se va nota şirul de date w şi şirurile de date c. Se va compara fiecare dintre şirurile de date c obținute astfel cu şirul de date w. Se va stabili care dintre cele două alegeri de parametri a fost mai inspirată. Şi comentariile inspirate de acest experiment vor fi consemnate în fişierul Word cu rezultatele lucrării amintit mai sus.

Lucrarea nr.2.

Metoda de criptare DES

1. Introducere

Este o metodă de criptare cu chei secrete. Pentru criptarea unui anumit mesaj se foloseşte o cheie secretă. Pentru decriptarea mesajului cifrat obţinut astfel se foloseşte aceeaşi cheie. Algoritmul DES (Data Encription Standard) este un algoritm de criptare cu chei simetrice fiind cel mai răspândit algoritm în întreaga lume. A fost adoptat de National Security Agency (NSA) din S.U.A. ca un standard de criptare.

Structural este constituit ca o combinaţie de algoritmi de tip transpoziţie şi subsituţie. El este construit pentru a cifra şi descifra blocuri de 64 biţi prin intermediul unei chei de criptare de 56 biţi.

2. Calcularea cheii

2.1 Se ia o cheie de 64 de biţi.

2.2 Se aplică regula bitului de paritate: fiecare octet din cheie trebuie să aibă un număr impar de biţi "1". Dacă un octet are număr par de biţi "1", atunci ultimul bit din octet se setează corespunzător.

2.3 Algoritmul de calculare a cheii

2.3.1 Se face permutarea următoare a cheii de 64 biţi (tot al 8-lea fiind scos, reducând astfel cheia la 56 biţi) folosind tabelul PC1. Bitul 1 (bitul cel mai semnificativ, MSB) al cheii permutate K+ este bitul 57 al cheii originale K, al 2-lea este bitul 49, ş.a.m.d, ultimul, adică bitul 56 din K+ fiind bitul 4 din K.

2.3.2 Se împarte cheia permutată, K+, în două jumătăţi. Primii 28 de biţi se notează cu C[0] şi ultimii 28 cu D[0].

2.3.3 Se calculează 16 subchei. Se începe cu i=1.

2.3.3.1 Se efectuează deplasări circulare spre stânga atât asupra lui C[i-1] cât şi asupra lui D[i-1], pentru a obţine pe C[i] respectiv D[i]. Numărul de permutări din fiecare iteraţie este dat de tabelul LS.

2.3.3.2 Se permută şirul concatenat C[i]D[i]după tabelul PC2. Din această permutare se vor obţine cheile K[i], care au o lungime de 48 de biţi.

2.3.3.3 Se reia ciclul de la **2.3.3.1** până este calculalt şi K[16].

Observaţie: Această etapă poate fi reprezentată grafic ca în figura următoare.

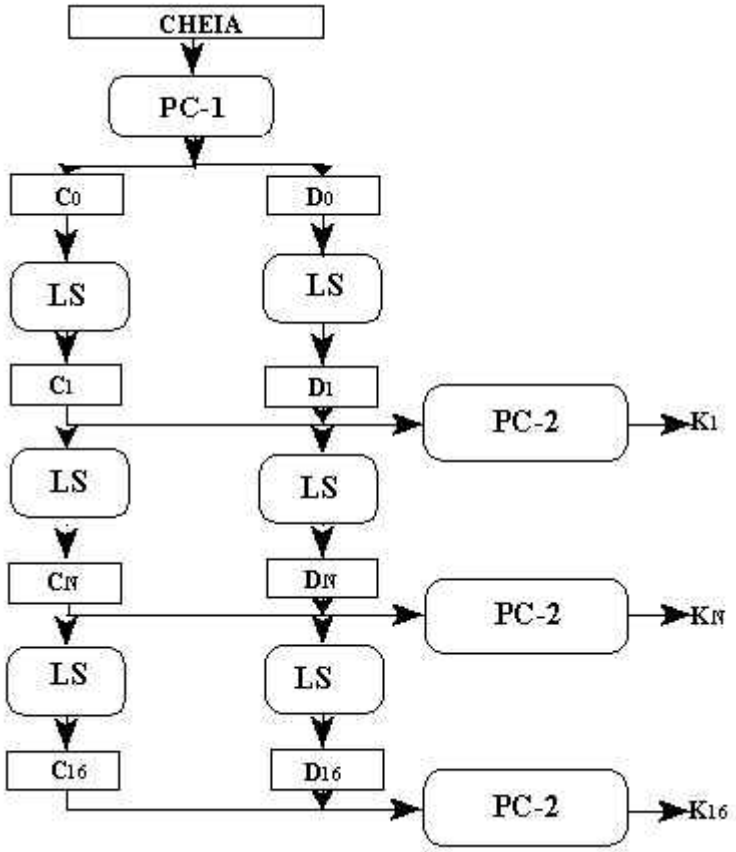


Figura 1. Mecanismul de generare a cheilor de iterație.

3. Criptarea blocului de date de 64 de biți

3.1 Se ia un bloc de date de 64 de biți. Dacă blocul este mai scurt, atunci se va completa cu zerouri în limita celor 64 de biți.

3.2 Se efectuează o permutare asupra blocului de date conform tabelului IP.

3.3 Se împarte blocul în două jumătăți. Primii 32 de biți se notează cu L[0], iar ultimii cu R[0].

3.4.Se aplică cele 16 subchei blocului de date. Se începe cu i=1.

3.4.1 Se extinde şirul R[i-1] de 32 de biți la 48 de biți după tabelul E.

3.4.2 Se efectuează operația XOR (SAU EXCLUSIV sau altfel spus adunare bit cu bit modulo 2) între şirul obținut anterior E(R[i-1]) şi cheia corespunzătoare, K[i].

3.4.3 Se împarte rezultatul operației de mai sus (E(R[i-1]) XOR K[i]) în 8 grupuri de 6 biți. Biții de la 1 la 6 se notează cu B[1], biții de la 7 la 12 cu B[2], ş.a.m.d., biții 43-28 fiind notați cu B[8].

3.4.4 Se înlocuiesc valorile găsite în «cutiile S» pentru toate grupurile B[j], începând cu j=1.; Fiecare valoare din «cutiile S» se consideră ca având 4 biți.

3.4.4.1 Se iau bitul 1 şi bitul 6 din grupurile B[j] împreună, ca un număr de 2 biți (să-i spunem m), care va indica rândul din cutia S[j] corespunzătoare grupului B[j].

3.4.4.2 Se iau biții 2, 3, 4 şi 5 împreună, ca un număr de 4 biți (să-l notăm cu n), care va indica coloana din cutia S[j] corespunzătoare grupului B[j].

3.4.4.3 Se înlocuieşte grupul B[j] cu numărul de 4 biți S[j][m][n].

3.4.4.4 Se va relula ciclul de la 3.4.4.1 până când toate cele 8 grupuri vor fi înlocuite.

3.4.5 Noile grupuri B[1] la B[8] se vor concatena (se vor pune împreună) după care asupra noului şir se va efectua o permutare în funcție de tabelul P.

3.4.6 Se face XOR între rezultatul etapei de mai sus şi L[i-1] pentru a-l obține pe R[i]. Toate aceste etape se pot scrie astfel:

R[i] = L[i-1] XOR P(S[1](B[]1)...S[8](B[8])),

unde B[j] este un grup de 6 biți obținut din E(R[i-1]) XOR K[i]. (Funcția după care se calculează R[i] poate fi scrisă mai concis astfel: R[i] = L[i-1] XOR f(R[i-1], K[i])).

Observație: Calculul funcției f se face ca în figura următoare:

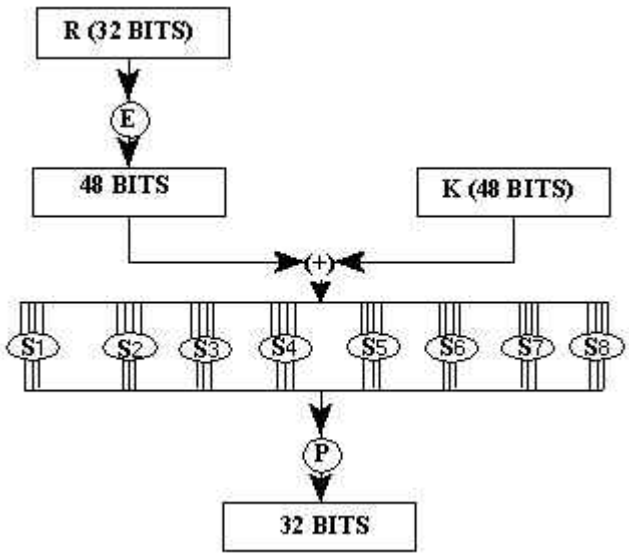


Figura 2. Calculul funcției f.

3.4.7 Se face înlocuirea L[i] = R[i-1]

3.4.8 Se reia ciclul de la **3.4.1** până când şi ultima cheie, adică K[16], a fost

aplicată blocului de date.

3.5 Se face permutarea asupra şirului R[16]L[16] (a se observa că de data asta, şirul R[16] este pus înaintea lui L[16]) conform tabelului IP^{-1}.

Observaţie: Calcularea blocului de date poate fi reprezentată grafic ca în figura următoare.

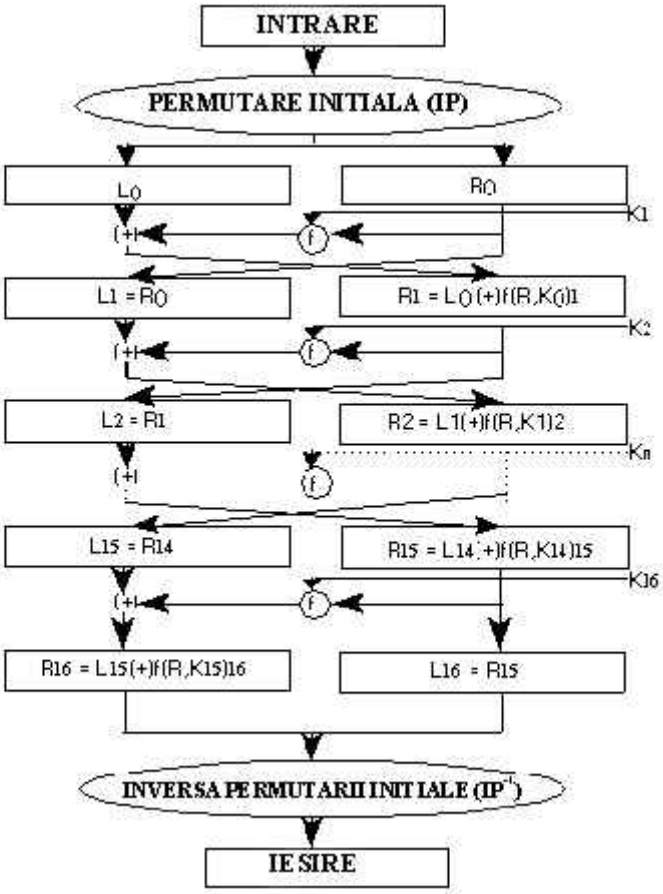


Figura 3. Criptarea unui mesaj.

4. Decriptarea unui bloc de date de 64 de biţi

4.1 Se vor repeta etapele de la 2 cu observaţia că subcheile K[i] se vor aplica în ordine inversă, adică se va începe cu cheia K[16], apoi K[15], K[14], ş.a.m.d. până la K[1].

5. Un exemplu de aplicare a algoritmului DES

Fie un mesaj M = 0123456789ABCDEF ce trebuie cripatat cu DES. M este un text în format hexazecimal. Rescriind pe M în format binar, vom obţine un bloc de 64 de biţi:
M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
pe care îl împărţim în două jumătăţi:
L = 0000 0001 0010 0011 0100 0101 0110 0111
R = 1000 1001 1010 1011 1100 1101 1110 1111
Atenţie! Atât M cât şi L şi R se pot scrie sub formă de matrice, acest lucru neinfluenţând deloc rezultatul criptării.
Vom citi de la stânga la dreapta astfel că primul bit din M este 0 ultimul fiind 1.
Algoritmul DES lucrează cu blocuri de text folosind chei de 56 de biţi. În realitate cheile apar ca având 64 de biţi dar fiecare al 8-lea bit din cheie nu este folosit (adică biţii de pe poziţile 8,

16, 24, 32, 40, 48, 56 şi 64). În orice caz noi vom scrie toţi cei 64 de biţi, numărându-i de la stânga la dreapta, în calculele ce vor urma, dar cum se va vedea dealtfel, biţii menţionaţi vor fi eliminaţi când vom crea cele 16 subchei.
Deci, să considerăm K, în forma hexazecimală, K = 133457799BBCDFF1 care să fie cheia cu care să criptăm mesajul M. Din nou, o vom tansforma în binar (1 = 0001, 3 = 0011. etc. şi se grupează în grupuri de câte 8, pentru a putea urmări mai uşor biţii care vor fi eliminaţi):

K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

Pasul 1: Crearea celor 16 subchei

Cheia K (64 de biţi) va fi permutată după tabelul PC-1 în modul următor: pentru că 57 este primul număr din tabel, înseamnă că al 57-lea bit din cheia K va deveni primul bit din cheia permutată K+; următorul număr din tabel este 49 deci al 49-lea bit din K va deveni al doilea bit din K+; ş.a.m.d.

Observaţie: Numai 56 de biţi din cheia iniţială vor fi permutaţi, adică nu vor fi folosiţi biţii cu indicii 8, 16, 24, 32, 40, 48, 56 şi 64.
Deci, avem
K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001
şi după permutare vom avea:
K+ = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111
În continuare, vom împărţi pe K+ în două jumătăţi C_0 şi D_0.
C_0 = 1111000 0110011 0010101 0101111
D_0 = 0101010 1011001 1001111 0001111
cu C_0 şi D_0 astfel obţinuţi vom crea cele 16 blocuri C_n şi D_n, 1<=n<=16. Fiecare pereche C_n şi D_n se va forma din perechea precedentă, C_{n-1} şi D_{n-1} prin deplasări spre stânga după tabelul LS. Pentru a efectual o deplasare spre stânga, se mută fiecare bit cu un loc spre stânga cu excepţia primului bit care este pus la sfârşitul blocului.

PC1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

LS

Numărul iterației	Numărul de deplasări spre stânga
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Asta însemnă că, de exemplu, C_3 şi D_3, se obțin din C_2 şi D_2, respectiv prin 2 deplasări spre stânga, iar C_{16} şi D_{16} se obțin din C_{15} şi D_{15} respectiv printr-o deplasare spre stânga a biților cu o poziție.
Astfel, vom obține:
C0 = 11110000110011001010101111
D0 = 01010101011001100111100011111
C1 = 11100001100110010101010111111
D1 = 10101010110011001111100011110
C2 = 11000011001100101010101111111
D2 = 01010101100110011111000111101
C3 = 00001100110010101010111111111
D3 = 01010110011001111100011110101
C4 = 00110011001010101011111111100
D4 = 01011001100111100011110101010
C5 = 11001100101010101111111110000
D5 = 01100110011110000111101010101
C6 = 00110010101010111111111000011
D6 = 10011001111000111101010101011
C7 = 11001010101011111111100001100
D7 = 01100111100011110101010110011
C8 = 00101010101111111110000110011
D8 = 10011110001111010101010110011
C9 = 01010101011111111100001100110
D9 = 00111100011110101010110011

C10 = 0101010111111110000110011001
D10 = 1111000111101010101011001100
C11 = 0101011111111000011001100101
D11 = 1100011110101010101100110011
C12 = 0101111111100001100110010101
D12 = 0001111010101010110011001111
C13 = 0111111110000110011001010101
D13 = 0111101010101011001100111100
C14 = 1111111000011001100101010101
D14 = 1110101010101100110011110001
C15 = 1111100001100110010101010111
D15 = 1010101010110011001111000111
C16 = 1111000011001100101010101111
D16 = 0101010101100110011110001111

În continuare vom forma cheile K_n, 1<=n<=16, aplicând tabelul PC-2 fiecărei perechi concatenate C_nD_n.
Observație: fiecare pereche concatenată are 56 de biți, dar PC-2 nu foloseşte decât 48 dintre aceştia.

PC2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Permutarea se va face exact ca şi în cazul lui PC-1, adică primul bit din Kn este bitul 14 din C_nD_n, al doilea este bitul 17 din C_nD_n, încheindu-se cu ce de-al 48-lea care este bitul 32 din C_nD_n.
Pentru prima cheie avem C_1D_1 = 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110
care după efectuarea permutării după PC-2 va deveni:
K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
La fel se procedează şi pentru celelalte 15 perechi şi vom obține în final:
K_2 = 011110 011010 111011 011001 110110 111100 100111 100101
K_3 = 010101 011111 110010 001010 010000 101100 111110 011001
K_4 = 011100 101010 110111 010110 110110 110011 010100 011101
K_5 = 011111 001110 110000 000000 111111 110101 001110 101000
K_6 = 011000 111010 010100 100111 110010 000111 101100 101111
K_7 = 111011 001000 010010 010110 111111 100001 100010 111100

K_8 = 111101 111000 101000 000111 010110 010011 101111 111011
K_9 = 111000 001101 101111 111101 011111 011110 011110 000001
K_{10} = 101100 011111 001101 101000 111101 100100 011001 001111
K_{11} = 001000 010101 111111 010011 110111 101101 001110 000110
K_{12} = 011101 010111 000111 110101 100101 000110 011111 101001
K_{13} = 100101 111100 010111 010001 111110 101011 101001 000001
K_{14} = 010111 110100 001110 110111 111100 101110 011100 111010
K_{15} = 101111 111001 000110 001101 001111 010011 111100 001010
K_{16} = 110010 110011 110110 001011 000011 100001 011111 110101
$K_1 - K_{16}$ sunt cele 16 subchei pe care le vom folosi pentru criptarea mesajului M.

Pasul 2: Critarea blocului de 64 de biți

Se va rearanja blocul inițial de text M după tabelul IP. Procedeul de permutare este acelaşi ca şi cel descris la PC-1 şi PC-2.

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
IP = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010
Pe IP îl vom împărți în două jumătăți L_0 şi R_0.
L_0 = 1100 1100 0000 0000 1100 1100 1111 1111
R_0 = 1111 0000 1010 1010 1111 0000 1010 1010
În continuare, vom face 16 iterații, 1<=n<=16, folosind funcția f, care având ca parametri două şiruri (un bloc de date de 32 de biți şi o cheie K_n de 48 de biți) va da ca rezultat un şir de 32 de biți.
Observație: în cele ce urmează se va considera semnul «+» ca reprezentând operația SAU Exclusiv XOR (adică adunare bit cu bit modulo 2)
Vom face următoarele calcule:
$L_n = R_{n-1}$
$R_n = L_{n-1} + f(R_{n-1}, K_n)$
Pentru n=1 avem:
K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
L_1 = R0 = 1111 0000 1010 1010 1111 0000 1010 1010
R_1 = L0 + f(R0,K1)
Să vedem cum aplicăm funcția f.
În primul rând îl vom extinde pe R_{n-1} folosind tabelul E.
Observație: tabelul E conține o serie de numere care se repetă astfel că face posibil ca dintr-un şir de 32 de biți să obținem unul de 48.

E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Pentru exemplificare să calculăm pe $E(R_0)$ din R_0.
R_0 = 1111 0000 1010 1010 1111 0000 1010 1010
$E(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101
Observație: fiecare grup de 4 biți din R_0 a fost extins într-unul de 6 biți.
Apoi, în calcularea funcției f, facem XOR între $E(R_{n-1})$ şi K_n: $K_n + E(R_{n-1})$
Astfel că pentru K_1 şi $E(R_0)$ vom avea:
K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
$E(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101
$K_1+E(R_0)$ = 011000 010001 011110 111010 100001 100110 010100 100111
Mai este un lucru de făcut în calcularea funcției f. În momentul de față avem 16 blocuri de 48 de biți pe care le-am aranjat fiecare în 8 grupuri de câte 6 biți fiecare. Acum vom folosi aşa numite «cutii S» (*S boxes*). Fiecărui grup de 6 biți i se va asocia, în ordine, o cutie S din cele 8.
Observație: fiecare bloc este împărțit în 8 grupuri, deci câte un grup pentru fiecare cutie.
Fiecare grup de 6 biți reprezintă o adresă în cutia care i-a fost asociată. La adresa indicată de grup vom găsi în cutie un număr de 4 biți care va înlocui grupul de 6 biți care l-a ales. Astfel că înlocuind toate cele 8 grupuri dintr-un bloc va rezulta un bloc de 32 de biți. Aceste operații le vom scrie astfel:

$K_n + E(R_{n-1}) = B_1B_2B_3B_4B_5B_6B_7B_8$
$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$

După cum spuneam mai înainte, fiecare grup de 6 biți reprezintă o adresă. Selecția numărului aflat în cutie la acea adresă se face în felul următor: se ia prima şi ultima cifră din grupul de 6 biți care va fi un număr binar de 2 biți, care reprezintă numărul rândului din cutie. Celelalte 4, care formează un număr de 4 biți reprezintă, trasnformat în zecimal, numărul coloanei din cutie. Spre exemplu, dacă avem grupul 010111 atunci numărul care îl vom selecta din cutia corespunzătoare se va afla pe rândul 1 (01) şi coloana 11 (adică 1011).

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2

6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Deci, pentru primul bloc avem:
K_1+E(R_0) = 011000 010001 011110 111010 100001 100110 010100 100111
$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 0101 1100 1000 0010 1011 0101 1001 0111
Ultimul pas în calcularea funcției f este să facem o permutare a rezulatului obținut din cutiile S.
f = $P(S_1(B_1)S_2(B_2)...S_8(B_8))$
Aceasta permutare se efectuează la fel ca şi Pc-1, PC-2, IP, etc.
P

16	**7**	**20**	**21**
29	**12**	**28**	**17**
1	**15**	**23**	**26**
5	**18**	**31**	**10**
2	**8**	**24**	**14**
32	**27**	**3**	**9**
19	**13**	**30**	**6**
22	**11**	**4**	**25**

În cazul nostru:
$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 0101 1100 1000 0010 1011 0101 1001 0111
f = 0010 0011 0100 1010 1010 1001 1011 1011
Având funcția f, putem să aplicăm formula:
R1 = L0 + f(R0,K1)
= 1100 1100 0000 0000 1100 1100 1111 1111
+ 0010 0011 0100 1010 1010 1001 1011 1011
= 1110 1111 0100 1010 0110 0101 0100 0100
Următorul pas va fi să facem $L_2 = R_1$ pe care l-am calculat mai înainte şi apoi să-l calculăm pe R_2 la fel ca mai înainte. Acest set de operații se va repeta de 16 ori, până vom avea determinate L_{16} şi $R_{16,}$ după care le punem împreună dar în ordine inversă, adică $R_{16}L_{16}$ şi aplicăm permutarea finală IP^{-1}.

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29

36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Fără a mai scrie calculele de la fiecare etapă, vom scrie direct pe L_{16} şi R_{16}
L_{16} = 0100 0011 0100 0010 0011 0010 0011 0100
R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101
$R_{16}L_{16}$ = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100
IP^{-1} = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101
care în hexa este 85E813540F0AB405
Aceasta este forma criptată a mesajului M = 0123456789ABCDEF cu cheia K = 133457799BBCDFF1.

5. Prezentarea programului folosit

Metoda de criptare DES este simulată cu ajutorul unui program scris în limbajul C. Lansarea acestui program se efectuează din Windows Commander făcând *click* pe DES.exe. Pe ecran apare o fereastră în care trebuie introdusă parola tti. Pe ecran apare un panou virtual de sistem de criptare DES, cum este cel din figura 4.

Figura 4. Panoul frontal al sistemului virtual de criptare DES.

Principalele funcţii ale acestui sistem sunt:
Teorie - se prezintă chestiunile teoretice conţinute în paragrafele anterioare (2 şi 3, respectiv 4) ale acestei lucrări. Deoarece fişierul de prezentare este de tip .html pot fi folosite la citire facilităţile specifice căutătoarelor pe INTERNET. O variantă conţinând mai puţine greşeli de ortografie poate fi citită, folosind programul Word, în fişierul Des.htm .
Făcând *click* pe butonul Teorie de pe panoul frontal, se activează fereastra prezentată în figura 5.

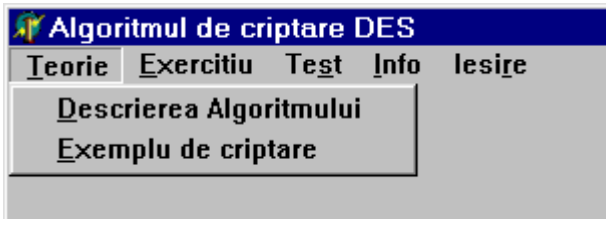


Figura 5. Subfuncţiile corespunzătoare funcţiei Teorie.

Făcând *click* pe butonul Descrierea Algooritmului, pe ecranul panoului frontal poate fi citit textul prezentat în paragrafele 2 şi 3. Făcând *click* pe butonul Exemplu de criptare, poate fi citit textul din paragraful 4. Pe durata citirii unuia dintre aceste texte celelalte funcţii ale sistemului virtual de criptare DES sunt dezactivate.

Exerciţiu. Făcând *click* pe butonul Exerciţiu, pe ecran apare fereastra prezentată în figura următoare.

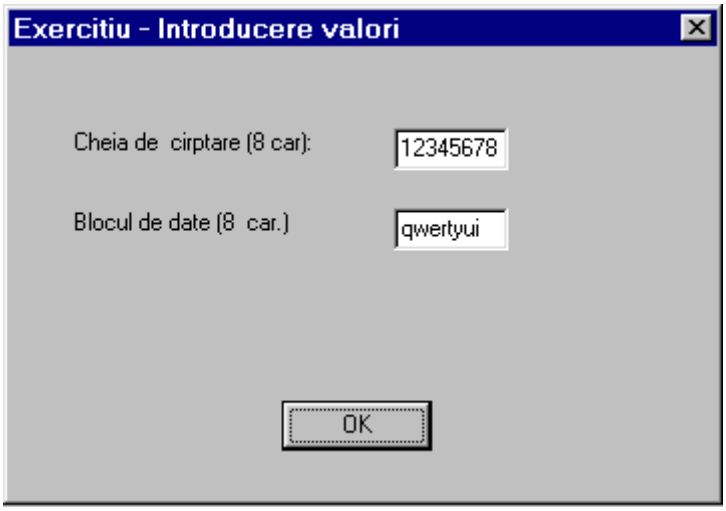


Figura 6. Subfuncţiile corespunzătoare funcţiei Exerciţiu.

Introducând 8 caractere pentru cheia de criptare şi 8 caractere pentru şirul de date se activează butonul OK din fereastra din figura anterioară. Făcând *click* pe acesta se trece la primul pas al algoritmului de criptare DES, în care cheia şi mesajul sunt transformate în şiruri de date cu lungimea de 64 de biţi. Algoritmul poate continua făcând *click* pe butonul > din noua fereastră afişată pe ecran. În pasul doi se generează cele 16 subchei de iteraţie. Valorile acestora sunt afişate în noua fereastră care apare pe ecran. Pasul al treilea corespunde criptării propriuzise. Este afişat pe ecran mesajul criptat. Ultimul pas corespunde decriptării. Se poate sesiza identitatea dintre textul în clar al mesajului iniţial şi textul obţinut în urma decriptării.

Info. Făcând *click* pe butonul Info se poate afla de ce parola acestui program este tti. Autorul acestui program l-a creat în urma frecventării unui cerc ştiinţific organizat de d-na Profesor dr. ing. Miranda Naforniţă.

Test. Făcând *click* pe butonul Test al panoului frontal din figura 4, poate fi făcută testarea cunoştiinţelor pe care le-aţi acumulat despre algoritmul DES.

Ieşire. Făcând *click* pe butonul Ieşire al panoului frontal din figura 4 se poate părăsi programul.

6. Desfăşurarea lucrării

6.1. Se verifică toate funcţiile programului evidenţiate în paragraful anterior. În acest mod se învaţă să se lucreze cu programul.
6.2. Se efectuează verificarea funcţionării programului efectuând o criptare şi o decripate, cu ajutorul opţiunii Exerciţiu. După efectuarea criptării se notează textul în clar (în cele trei

forme ale sale: şir de caractere, binar, hexazecimal) şi şirul de date obținut în urma criptării. Acesta se foloseşte la decriptare, urmând ca după aceasta să se regăsească textul în clar folosit la criptare. Mesajul în clar şi criptat vor fi salvate într-un fişier Word, denumit cu numele utilizatorului şi indicele lucrării de laborator, pentru a putea fi identificat ulterior, în directorul Users.

6.3. Pentru acelaşi mesaj se va simula folosirea a două sisteme de criptare DES, prin alegerea, a două chei inițiale diferite. Pentru fiecare dintre aceste experimente se vor nota formele în clar şi criptată ale mesajului. Se vor compara cele două mesaje criptate obținute astfel numărându-se caracterele comune (care se găsesc în acelaşi loc şi au aceaşi valoare). Pe baza acestui criteriu se va stabili care dintre cele două alegeri de parametri a fost mai inspirată. Şi comentariile inspirate de acest experiment vor fi consemnate în fişierul Word cu rezultatele lucrării amintit mai sus.
6.4. Se va efectua testul propus alegînd opțiunea Test. Dacă nu se obține un rezultat satisfăcător testul se reia după ce se recitesc în prealabil paragrafele 2,3 şi 4 ale acestui document.

Lucrarea 3.

O utilizare posibilă a parolelor, protecția unui document

1. Introducere

Procesorul de text, Word, propune mai multe mijloace de limitare a accesului la un document, bazate pe folosirea unei parole. Este posibilă:

- Atribuirea unei parole pentru deschiderea documentului în scopul împiedicării deschiderii sale de către utilizatori neautorizați,
- Atribuirea unei parole în scopul împiedicării utilizatorilor (care nu o cunosc) de a efectua modificări, (ei pot însă să deschidă documentul). Dacă un utilizator deschide documentul fără a folosi parola şi apoi îl modifică, el va trebui să salveze noul document creat cu un alt nume;
- Se recomandă deschiderea documentului doar pentru citire. Dacă un utilizator deschide documentul doar pentru citire şi îl modifică apoi, el trebuie să dea un nou nume documentului nou creat, pentru a-l putea înregistra. În schimb dacă un utilizator deschide documentul în modul citire-scriere şi îl modifică, atunci el îl poate înregistra sub acelaşi nume.
- Se recomandă atribuirea unei parole cu ocazia distribuirii unui document, pentru a împiedica orice modificare a acestuia, cu excepția comentarilor sau a corecturilor.
- Se recomandă folosirea unei parole atunci când se crează un formular cu ajutorul câmpurilor de formular, pentru a împiedica utilizatorii de a modifica secțiunile specificate.

Atenție. Dacă ați uitat parola pe care ați atribuit-o unui document nu veți mai putea să deschideți acest document, să-i suprimați protecția sau să recuperați datele conținute în acesta. De aceea se recomandă să se păstreze într-un loc sigur o listă a parolelor şi a documentelor corespunzătoare.

1.1. Parolă pentru deschiderea unui document

Se deschide documentul. În meniul Fichier (File) se face *click* pe Enregistrer sous, (Save as), iar apoi se face *click* pe Options. În zona Mot de passe pour la lecture (Password for reading) se tastează o parolă şi apoi se face *click* pe OK. În final se face *click* pe Enregistrer (Save). Următoarea deschidere a acestui fişier va putea fi făcută doar după ce se comunică parola corectă.

1.2. Parolă pentru modificarea unui document

Se deschide documentul. În meniul Fichier (File) se face *click* pe Enregistrer sous, (Save as), iar apoi se face *click* pe Options. În zona Mot de passe pour la modification (Password for modification) se tastează o parolă şi apoi se face *click* pe OK. Apoi se face *click* pe Enregistrer (Save).

Sfat. Dacă se distribuie un document pentru lectură li se poate permite lectorilor acestuia să emită sugestii cu ajutorul câmpurilor Commentaires (Adnotations) sau Revisions, protejând documentul pentru orice alt tip de modificare. În meniul Outils (Tools), faceți *click* pe Protejer le document (Protect the document) şi apoi pe opțiunea Revisions. Pentru a

permite recenzorilor să insereze comentarii fără a modifica conținutul documentului faceți *click* pe opțiunea Commentaires (Adnotations).

1.3. Deschiderea documentului doar în modul citire

Se deschide documentul. Se apasă, în meniul Fichier (File) pe Enregistrer sous (Save as). Se apasă apoi pe Options. Apoi se activează câmpul Lecture seule recommandee (Read only) şi apoi se apasă pe OK. În final se apasă pe Enregistrer (Save).

1.4. Pregătirea unui document pentru verificare

Utilitarul Microsoft Word pune la dispoziție mai multe opțiuni de trimitere a unui document la verificare şi câteva opțiuni de urmărire a modificărilor efectuate, de încorporare şi de conservare a unei înregistrări a modificărilor aduse.

În cazul în care se dorește revizuirea unui document de către o echipă, dar se dorește ca decizia finală, legată de acceptarea sau refuzarea unei modificări, să fie luată de o singură persoană, pot fi pregătite mai multe copii ale documentului care să fie distribuite membrilor echipei de revizuire, care-şi consemnează observațiile în mod electronic.

Pentru a asigura urmărirea modificărilor, utilitarul Word foloseşte opțiunea Marques de revision pentru a putea fi văzute schimbările şi comentariile fiecărui membru al echipei de îndată ce acesta a încheiat sarcina sa. În acest fel persoana responsabilă poate decide care modificări şi comentarii trebuie incluse în forma finală a documentului.

Dacă se dorește includerea ulterioară a altor membri în echipa de corectare, documentul inițial poate fi configurat în aşa fel încât Word să salveze automat o variantă instantanee a fiecărei copii realizate de către fiecare corector atunci când acesta închide documentul respectiv, după ce l-a modificat. Toate versiunile sunt stocate în acelaşi document, dar în modul de lucru normal programul Word afişează doar versiunea curentă. Pentru fiecare versiune a documentului programul Word înregistrează data şi ora salvării precum şi numele persoanei care a efectuat acele modificări. Orice versiune anterioară poate fi afişată într-o fereastră distinctă deschizând-o cu ajutorul cutiei de dialog, Versions.

1.5. Protejarea textului fix al unui document

După creerea unui formular se poate ca acesta să fie protejat astfel încât utilizatorii săi să nu poată completa decât în anumite zone. Pentru aceasta în meniul Outils (Tools), se face *click* pe Protection du document (Document protection). Apoi se face *click* pe Formulaire (Form). Apoi se tastează o parolă în zona Mot de passe (facultatif) (Password). Şi utilizatorii care nu cunosc această parolă pot să completeze formularul. Pentru a proteja integritatea formularului se face *click* pe OK. Pentru a proteja doar anumite părți ale formularului e necesar ca acestea să se găsească în secțiuni distincte. De aceea trebuie făcut *click* pe Sections şi apoi trebuie dezactivată opțiunea de secțiuni pentru cele care nu se dorește a fi protejate.

Sfat. Pe durata creării sau a modificării unui formular protecția sa poate fi activată sau dezactivată rapid, făcând *click* în meniul Formulaires pe Protection du formulaire.

2. Desfăşurarea lucrării

Scopul acestei lucrări este familiarizarea cu sistemul de protecție prin parole al documentelor, specific utilitarului Word, din pachetul de programe Office, conceput la firma Microsoft. Se exemplifică toate posibilitățile de protecție prin parole descrise mai sus.

2.1. Se porneşte programul Word. Se crează un document, conţinînd minimul două propoziţii. Apoi în meniul Fichier (File) se face *click* pe Enregistrer sous, (Save as), iar apoi se face *click* pe Options. În zona Mot de passe pour la lecture (Password for reading) se tastează o parolă şi apoi se face *click* pe OK. În final se face *click* pe Enregistrer (Save). Se încearcă deschiderea aceluiaşi document. Ce se constată ? Răspunsul la această întrebare va fi consemnat într-un nou document, care va conţine un referat despre această lucrare şi care va fi salvat într-un fişier al cărui nume va conţine numele dumneavoastră şi numărul de ordine al acestei lucrări, într-un director care poartă numele dumneavoastră, situat în directorul Users, de pe calculatorul pe care lucraţi. În acelaşi fişier va fi salvat şi documentul creat anterior şi parolat.

2.2. Se crează un nou document, care va conţine de asemenea cel puţin două propoziţii. În meniul Fichier (File) se face *click* pe Enregistrer sous, (Save as), iar apoi se face *click* pe Options. În zona Mot de passe pour la modification (Password for modification) se tastează o parolă şi apoi se face *click* pe OK. Apoi se face *click* pe Enregistrer (Save). Apoi se încearcă deschiderea documentului salvat anterior. Ce se constată ? Răspunsul la această întrebare, consemnat în documentul referat împreună cu cel de al doilea fişier parolat se salvează în directorul din Users amintit mai sus.

2.3. Se crează un nou document care va conţine de asemenea cel puţin două propoziţii. Apoi se face *click* pe Outils (Tools) din meniul principal al programului Word. Apoi se face *click* pe opţiunea Protection du document. În noua fereastră apărută se selectează opţiunea Modifications (Changes), iar în zona Mot de passe (faultatif), (Password), se înscrie o parolă. După aceea se face *click* pe butonul OK. Apare o nouă fereastră, în care se înscrie din nou aceeaşi parolă. Apoi se salvează documentul făcându-se *click* pe Fichier (File), apoi pe Enregistrer sous (Save as) şi în final pe Enregistrer (Save). Se redeschide documentul salvat anterior. Se încearcă ştergerea textului. Ce se constată ? Răspunsul la această întrebare va fi consemnat în documentul care conţine referatul acestei lucrări. Apoi se selectează opţiunea Outils (Tools) din meniul principal al programului Word. Se face *click* pe Oter le document şi în fereastra apărută se înscrie parola folosită. Apoi se încearcă din nou ştergerea unor caractere din document. Ce se constată de această dată ? Şi răspunsul la această întrebare va fi consemnat în referatul acestei lucrări. Se repetă experienţa de mai sus alegând pe rând celelalte două opţiuni din fereastra care apare după ce se face *click* pe Protection du document (Document protection): Commentaires (Adnotations) şi Formulaires (Forms). Când se alege opţiunea Commentaires (Adnotations) trebuie ca din meniul principal al programului Word să se aleagă opţiunea Insertion şi apoi Commentaires. În acest mod vor putea fi incluse comentarii în documentul considerat.

2.4. Imaginaţi un experiment pentru a verifica afirmaţiile din paragraful 1.5.

Lucrarea 4.

Criptarea rapidă a directoarelor şi fişierelor pentru transmiterea lor prin poştă electronică, folosind metoda de criptare IDEA

1. Introducere

Metodele de criptare cu cheie secretă, ca de exemplu DES, AES sau IDEA, pot fi utilizate pentru transmiterea sigură a unor fişiere prin poşta electronică. Un astfel de fişier poate fi criptat şi transmis ca şi document ataşat, cu ajutorul oricărui program de poştă electronică. Scopul acestei lucrări este familiarizarea cu un program de criptare rapidă a fişierelor care poate fi utilizat pentru transmisii prin poştă electronică. Acest program se numeşte DATAGUARD şi se bazează pe folosirea algoritmului IDEA.

2. Algoritmul IDEA

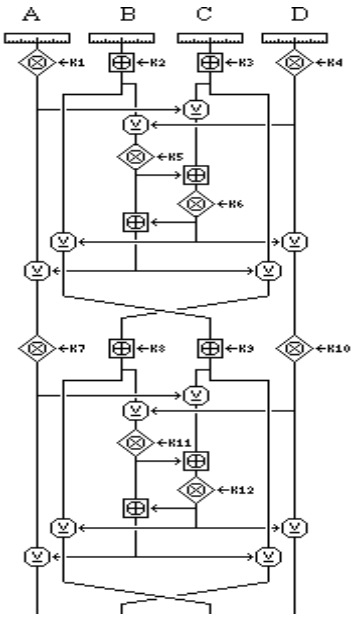
A fost conceput în Elveţia de către Xuejia Lai şi James Massey în anul 1992. Patentul său se găseşte la firma Ascom. Principala sa aplicaţie este programul de criptare pentru poşta electronică, PGP, *Prety Good Privacy*. Este unul dintre sistemele de criptare cele mai rapide şi mai sigure disponibile la ora actuală. Foloseşte o cheie de 128 de biţi. Pe baza acesteia se construiesc 52 de sub-chei cu lungimea de 16 biţi fiecare. Câte două dintre acestea se utilizează la fiecare dintre cele 8 iteraţii ale algoritmului, şi câte 4 se utilizează înaintea fiecărei iteraţii şi după ultima iteraţie. Nu foloseşte nici un tabel de alocare de biţi şi nici o cutie de tip S. Cifrarea şi descifrarea se fac pe blocuri de câte 64 de biţi. Se bazează pe utilizarea unor operaţii algebrice utile în operaţiile de criptare cum ar fi suma modulo 2, suma modulo 2^{16} , produsul modulo 2^{16} +1. În continuare se dau câteva explicaţii referitoare la aceste înmulţiri. Operaţia de înmulţire cu zero are ca rezultat zero şi nu este o operaţie inversabilă. Dar înmulţirea folosită în acest algoritm trebuie să fie o operaţie inversabilă. Numărul 2^{16} +1 are valoarea 65537 şi este prim. Pe baza tabelului de înmulţire specific mulţimii claselor de resturi modulo 2^{16} +1 se constată că această operaţie este inversabilă dacă se evită înmulţirea cu zero (se elimină din tabel prima linie şi prima coloană). Toate aceste operaţii se efectuează asupra unor sub-blocuri de 16 biţi. S-a dovedit că IDEA este mai sigur decât DES la atacuri de criptanaliză diferenţială.

2.1. Descrierea algoritmului IDEA

Fie cele patru sferturi ale textului clar care trebuie criptat notate cu A, B, C şi D, şi cele 52 de sub-chei notate cu K(1) … K(52).
Înainte sau în cursul primei iteraţii se efectuează următoarele operaţii:
Se înmulţeşte A cu K(1). Rezultatul va reprezenta noua valoare a lui A. Se adună modulo 2^{16} K(2) la B. Rezultatul va reprezenta noua valoare a lui B. Se adună modulo 2^{16} K(3) la C. Rezultatul va reprezenta noua valoare a lui C. Se înmulţeşte D cu K(4). Rezultatul va reprezenta noua valoare a lui D.
Prima iteraţie propriuzisă constă din executarea următoarelor operaţii:
Se calculează suma modulo 2 dintre A şi C (rezultatul se notează cu E). Se calculează suma modulo 2 dintre B şi D (şi se notează cu F). Se înmulţeşte E cu K(5). Rezultatul reprezintă noua valoare a lui E. Se adună modulo 2^{16} noua valoare a lui E la F. Rezultatul va reprezenta noua valoare a lui F. Se înmulţeşte noua valoare a lui F cu K(6). Se adună rezultatul, care reprezintă noua valoare a lui F, la E. Se modifică valorile lui A şi C însumând modulo 2 aceste valori cu valoarea curentă a lui E. Noile valori pentru A şi C se substituie una celeilalte

obținându-se astfel două dintre cele 4 blocuri inițiale ale celei de a doua iterații şi anume A şi C. Se modifică valorile lui B şi D, însumând modulo 2 aceste valori cu valoarea curentă a lui F. Noile valori pentru B şi D se substituie una celeilalte obținându-se astfel celelalte două dintre cele 4 blocuri inițiale ale celei de a doua iterații şi anume B şi D. Prima iterație este prezentată în figura 1. Celelalte 7 iterații sunt identice, doar că se folosesc celelalte sub-chei: de la K(7) la K(12) pentru cea de a doua iterație şi de la K(43) la K(48) pentru cea de a 8-a iterație. La ultima iterație nu se mai face substituția finală dintre A şi C respectiv B şi D.

Ultimele operații sunt: Se înmulțeşte A cu K(49). Se adună modulo 2^{16} K(50) la B. Se adună modulo 2^{16} K(51) la C. Se înmulțeşte D cu K(52). Cele 8 iterații ale algoritmului IDEA sunt prezentate în figura 2.



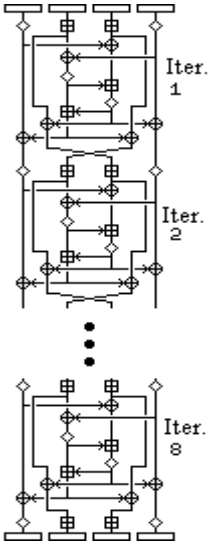


Figura 1. Prima iterație. Figura 2. Structura algoritmului IDEA.

2.2. Decriptarea

Cum se poate inversa o iterație a algoritmului IDEA, când toate cele 4 blocuri se modifică în acelaşi timp ? Răspunsul se bazează pe o proprietate a sumei modulo 2. Suma modulo 2 a două variabile A şi C nu se modifică atunci când cele două variabile sunt însumate modulo 2 cu o aceeaşi variabilă, X.

D:

$$A_n = A_v + X; \quad C_n = C_v + X; \quad \Rightarrow A_n + C_n = A_v + C_v + X + X = A_v + C_v$$

Se observă că variabila X a dispărut. Aceeaşi proprietate este valabilă şi pentru variabilele B şi D. Întrucât variabilele folosite în iterațiile algoritmului IDEA sunt funcții de $A + C$ şi $B + D$ rezultă că cele 4 variabile pot fi recuperate. Inversarea operației de adunare modulo 2^{16} se face prin calculul complementului față de 2. Pentru prima iterație a algoritmului de decriptare se utilizează următoarele chei:

```
KD(1) = 1/K(49)
KD(2) =  -K(50)
KD(3) =  -K(51)
KD(4) = 1/K(52)
```

Pentru cheile din următoarele iteraţii procedura următoare se repetă de 8 ori, adunând 6 la fiecare indice al unei chei de decriptare şi scăzând 6 din fiecare indice al unei chei de criptare:

```
KD(5)   =    K(47)
KD(6)   =    K(48)

KD(7)   = 1/K(43)
KD(8)   =  -K(45)
KD(9)   =  -K(44)
KD(10)  = 1/K(46)
```

2.3. Generarea sub-cheilor

Primele 8 sub-chei se obţin prin segmentarea cheii originale a algoritmului IDEA în segmente de 16 biţi. Apoi se efectuează o deplasare circulară la stânga a cheii originale cu 25 de poziţii şi o nouă segmentare obţinându-se următoarele 8 chei. Această procedură de deplasare la stânga şi segmentare este repetată până când se obţin toate cele 52 de chei de criptare necesare.

3. Programul DATAGUARD

Realizează o criptare (decriptare) rapidă a fişierelor şi directoarelor. Datele criptate astfel pot fi transmise prin programe comune de poştă electronică în reţele publice (inclusiv INTERNET), asigurându-li-se securitatea. Utilizarea unor algoritmi eficienţi şi optimizaţi, reduce pierderile de performanţă ale sistemului de transmisiuni datorate proceselor de criptare şi de decriptare, fără a scădea securitatea comunicaţiei.

3.1. Managementul parolelor

O calitate remarcabilă a acestui program este autorizarea accesului utilizatorilor prin parole. Pot fi realizate mai multe clase de utilizatori, fiecare constând dintr-un număr oricât de mare de membri. În scopul decriptării unui anumit fişier toţi membrii unui anumit grup trebuie să-şi folosească parolele individuale. Fiecare membru poate aparţine la diferite grupuri în acelaşi timp (folosind aceaşi parolă). Există de asemenea posibilitatea ca oricare doi membri ai unui grup, format de exemplu din opt membri, să decripteze împreună un fişier (Principiul celor patru ochi).

Parolele pot fi introduse de la tastatură sau de pe purtătoare de date (de exemplu dischete). Nu este necesar ca parola de pe dischetă să fie recunoscută ca atare, orice fişier de pe dischetă poate fi folosit drept parolă (ca de exemplu un fişier de imagine, sau un fişier de sunet sau un fişier de text).

3.2. Algoritmi folosiţi

Algoritmul IDEA a dobândit certificatul ISO, ISO/IEC 9979 şi lucrează cu o cheie fixă de 128 de biţi, iar datele mesajului sunt grupate în blocuri de câte 64 de biţi.

Algoritmul SEAL(TM) criptează fiecare bit separat, fiind un sistem de criptare de secvenţă. Acest algoritm a fost conceput la IBM de către P. Rogaway şi D. Coppersmith. Lucrează cu o cheie de 160 de biţi. Acest algoritm a fost optimizat pentru procesoare de 32 de biţi şi se crede că este cel mai rapid şi mai sigur sistem de criptare soft existent pe piaţă asigurând performanţe ridicate pe durata criptării şi a decriptării, dovedind că ipoteza că sistemele de criptare hard sunt mai rapide decât sistemele de criptare soft este falsă. Dezavantajul acestui algoritm, în comparaţie cu algoritmii de criptare pe blocuri, ca de

exemplu IDEA, este că necesită o fază de inițializare înainte de a începe procedura de criptare sau decriptare a unui fişier. De aceea acest algoritm este mai lent decât algoritmul IDEA în cazul fişierelor de criptat scurte. Acesta este motivul pentru care în programul DATAGUARD există un comutator pentru selectarea celui mai potrivit algoritm pentru un anumit mesaj de criptat. La decriptare programul DATAGUARD recunoaşte automat algoritmul folosit pentru codare. Şi alți algoritmi de criptare pot fi incluşi în programul DATAGUARD la cererea clienților.

3.3. Varianta Demo folosită în această lucrare

Programul folosit în această lucrare este o variantă, Demo, a programului DATAGUARD. Varianta comercială a acestui program are un domeniu larg de utilizare. Unele funcții nu sunt disponibile în această variantă Demo.

- Tehnologia de criptare folosită în această variantă nu asigură acelaşi nivel de confidențialitate ca în cazul variantei comerciale. Se utilizează chei de 12 biți şi nu de 128 sau 160 de biți.
- Pot fi creați doar utilizatori diferiți, nu şi grupuri de utilizatori diferite.
- Programul de "Help" nu este complet.

Atenție.
Dacă uitați parola cu care ați criptat anumite fişiere acestea vor rămâne criptate (pe durata criptării fişierul original este ascuns) şi nu le veți mai putea folosi. Algoritmii din această variantă lucrează cu chei de lungime mai scurtă.
Nu folosiți parole simple. De securitatea acestora depinde securitatea criptării fişierelor. Nu folosiți drept parole nume, numere de telefon sau date de naştere.

3.3.1. Panoul frontal al programului DATAGUARD

Acest panou frontal vă permite să definiți controale sau să creați clase, utilizatori sau grupuri. Puteți, de asemenea, să criptați sau să decriptați, direct, folosind acest panou frontal. Puteți specifica dacă doriți ca programul DATAGUARD să se lanseze automat la pornirea sistemului. În acest scop trebuie marcat câmpul Auto Load de pe panoul frontal. Dacă prezența pe ecran a acestui panou frontal vă deranjează, apăsați butonul "Close" şi panoul frontal va dispărea. Veți continua să vedeți o iconiță DATAGUARD pe *task bar.*

3.3.1.1. Configurarea programului DATAGUARD

1. Definiți cel puțin o clasă.
2. După ce ați făcut asta, puteți defini utilizatorul (utilizatorii) care au acces la clasă (clase).

Asta este tot. Acum puteți experimenta programul DATAGUARD. Apăsați butonul *Encrypt* din colțul dreapta jos al panoului frontal…

3.3.1.1.1. Key management

Pot fi create, editate sau şterse clase. Pentru a efectua aceste acțiuni este suficient să se apese unul din butoanele din partea dreaptă. Numele claselor sunt afişate pe coloana din stânga. Pe coloana din dreapta pot fi văzuți utilizatorii sau grupurile din această clasă.

3.3.1.1.2. Password settings

Pot fi specificate parole pentru fiecare clasă. După ce ați făcut asta, le puteți secretiza cu ajutorul unei parole. Schimbările pe care le faceți se vor aplica doar la utilizatori noi.

3.3.1.1.3. Encryption settings

Poate fi specificat algoritmul de criptare pentru fiecare clasă şi puteți defini cum să manipuleze programul DATAGUARD fişierele criptate.

4. **Desfăşurarea lucrării**

Se instalează varianta Demo a programului DATAGUARD, făcând *click* pe DG32SETUP.EXE. Se configurează acest program, constituindu-se o clasă, formată din 2 utilizatori, unul fiind operatorul calculatorului respectiv iar celălalt operatorul unui calculator vecin. Aceaşi clasă va fi creată şi pe calculatorul vecin (cei doi utilizatori îşi vor alege pentru cele două calculatoare, aceeaşi parolă).

4.1. Cele două calculatoare îşi vor activa legătura prin INTRANET, între partițiile care conține directorul Users/STII. În acest scop din programul Windows Commander de pe fiecare calculator se va alege opțiunea Commands şi apoi Share Current Directory, apoi Sharing. Se va crea un fişier cu numele original.txt. Acesta va fi salvat în directorul Users/STII al calculatorului respectiv. Apoi va fi criptat folosind algoritmul IDEA şi parola utilizatorului calculatorului respectiv. Din directorul Users/STII va dispărea fişierul original.txt şi va apărea un fişier cu numele original.txt.Ctx. Acesta va putea fi decriptat folosind aceeaşi parolă, de către acelaşi utilizator. Se va verifica faptul că procesul de criptare-decriptare nu introduce erori.

4.2. Se va repeta experimentul descris mai sus pentru fişiere de tip .doc, .bmp şi .pdf.

4.3. Se va crea un fişier cu numele original.txt. Acesta va fi salvat în directorul Users/STII al calculatorului respectiv. Apoi va fi criptat folosind algoritmul IDEA şi parola utilizatorului respectiv. Din directorul Users/STII va dispărea fişierul original.txt şi va apărea un fişier cu numele original.txt.Ctx. Acest fişier se va transmite calculatorului omolog prin INTRANET în directorul Users/STII. Aici va fi decriptat folosind parola utilizatorului acelui calculator. În directorul Users/STII al celui de al doilea calculator va apărea fişierul original.txt . Se va verifica identitatea dintre fişierele original.txt, creat inițial şi fişierul obținut după decriptare.

4.4. Se va repeta experimentul descris mai sus pentru fişiere de tip .doc, .bmp şi .pdf.

4.5. Se vor repeta experimentele descrise mai sus folosindu-se, de această dată, algoritmul de criptare SEAL.

4.6. Dezinstalați programul DATAGUARD.

Lucrarea 5.

Metoda de criptare AES

1. Introducere

Recent a fost omologat un nou standard de criptare simetrică a datelor, *The Advanced Encryption Standard*, care urmează să înlocuiască vechiul standard, DES. Acest standard este bazat pe algoritmul Rijndael.

2. Algoritmul Rijndael

Algoritmul care a câştigat competiţia pentru standardul AES este numit Rijndeal. Acesta realizează doar operaţii pe octeţi întregi. El este foarte flexibil deoarece dimensiunea blocurilor cu care se lucrează poate fi aleasă de valoare 128, 192 sau 256 de biţi. Descrierea originală a algoritmului Rijndael se găseşte la adresa: http://www.esat.kuleuven.ac.be/~rijmen/rijndael/ .
În continuare se prezintă varianta care lucrează cu blocuri de 128 de biţi. Rijndael are un număr variabil de iteraţii. Acesta poate fi (fără a calcula ultima iteraţie care nu este completă):

- 9 dacă atât blocurile cât şi cheia iniţială au o lungime de 128 de biţi;
- 11 dacă fie blocurile fie cheia iniţială au lungimea de 192 de biţi şi nici una dintre ele nu are o lungime superioară acestei valori;
- 13 dacă atât blocurile cât şi cheia au o lungime de 256 de biţi.

Pentru a cripta un bloc de date cu algoritmul Rijndael, primul pas presupune calculul unui sau-exclusiv între blocul de text clar şi o sub-cheie. Paşii următori sunt iteraţiile care se vor prezenta în continuare. Ultimul pas este constituit de ultima iteraţie care este incompletă, neconţinând operaţia de amestecare a coloanelor, *the Mix Column step.*

2.1. Iteraţiile algoritmului

Fiecare iteraţie obişnuită se efectuează în 4 paşi. Primul pas este cel de substituire al octeţilor, *the Byte Sub step*. În acest pas fiecare octet al textului clar este substituit cu un octet extras dintr-o cutie de tip S. Cutia de tip S este descrisă de matricea:

99	124	119	123	242	107	111	197
48	1	103	43	254	215	171	118
202	130	201	125	250	89	71	240
173	212	162	175	156	164	114	192
183	253	147	38	54	63	247	204
52	165	229	241	113	216	49	21
4	199	35	195	24	150	5	154
7	18	128	226	235	39	178	117
9	131	44	26	27	110	90	160
82	59	214	179	41	227	47	132
83	209	0	237	32	252	177	91
106	203	190	57	74	76	88	207
208	239	170	251	67	77	51	133
69	249	2	127	80	60	159	168
81	163	64	143	146	157	56	245
188	182	218	33	16	255	243	210
205	12	19	236	95	151	68	23

196 167 126 61 100 93 25 115
96 129 79 220 34 42 144 136
70 238 184 20 222 94 11 219
224 50 58 10 73 6 36 92
194 211 172 98 145 149 228 121
231 200 55 109 141 213 78 169
108 86 244 234 101 122 174 8
186 120 37 46 28 166 180 198
232 221 116 31 75 189 139 138
112 62 181 102 72 3 246 14
97 53 87 185 134 193 29 158
225 248 152 17 105 217 142 148
155 30 135 233 206 85 40 223
140 161 137 13 191 230 66 104
65 153 45 15 176 84 187 22

Cel de al doilea pas al unei iterații uzuale se numeşte deplasarea liniilor, *the Shift Row step*. Considerând că blocul care trebuie construit este alcătuit cu octeții numerotați de la 1 la 16, aceşti octeți se aranjează într-un dreptunghi şi se deplasează
după cum urmează:

De la la
1 5 9 13 1 5 9 13
2 6 10 14 6 10 14 2
3 7 11 15 11 15 3 7
4 8 12 16 16 4 8 12

Cel de al treilea pas al algoritmului de criptare Rijndael este numit amestecarea coloanelor, *the Mix Column step*. Acest pas se realizează prin înmulțire matricială: fiecare coloană, în aranjamentul pe care l-am observat, este înmulțită cu matricea:

2 3 1 1
1 2 3 1
1 1 2 3
3 1 1 2

Această înmulțire matricială corepunde unei înmuțiri specifică câmpului Galois al lui 2^8, definită de polinomul modul $x^8 + x^4 + x^3 + x + 1$. Această înmulțire (folosind acelaşi polinom modul) a fost prezentată şi exemplificată în paragraful destinat bazelor matematice ale criptării. Octeții care trebuie înmulțiți sunt priviți ca şi polinoame şi nu ca şi numere. De exemplu prin înmulțirea unui octet cu 3 se obține rezultatul operației sau-exclusiv dintre acel octet şi şi varianta sa obținută prin rotirea acelui octet cu o poziție la stânga. Dacă rezultatul acestei înmulțiri are mai mult de 8 biți, biții suplimentari nu sunt pur şi simplu ignorați. Pentru eliminarea lor se calculează sau-exclusiv între rezultatul obținut (în urma "înmulțirii" deja efectuate) (deplasat la stânga dacă este necesar) şi şirul binar cu lungimea de 9 biți; 100011011 (care corespunde polinomului modul).
Cel de al patrulea pas al algoritmului Rijndael este cel de adăugare a sub-cheii, *the Add Round Key step.*

Acesta presupune doar calculul unui sau-exclusiv cu sub-cheia specifică iterației curente. O iterație uzuală a acestui algoritm are aspectul din figura următoare.

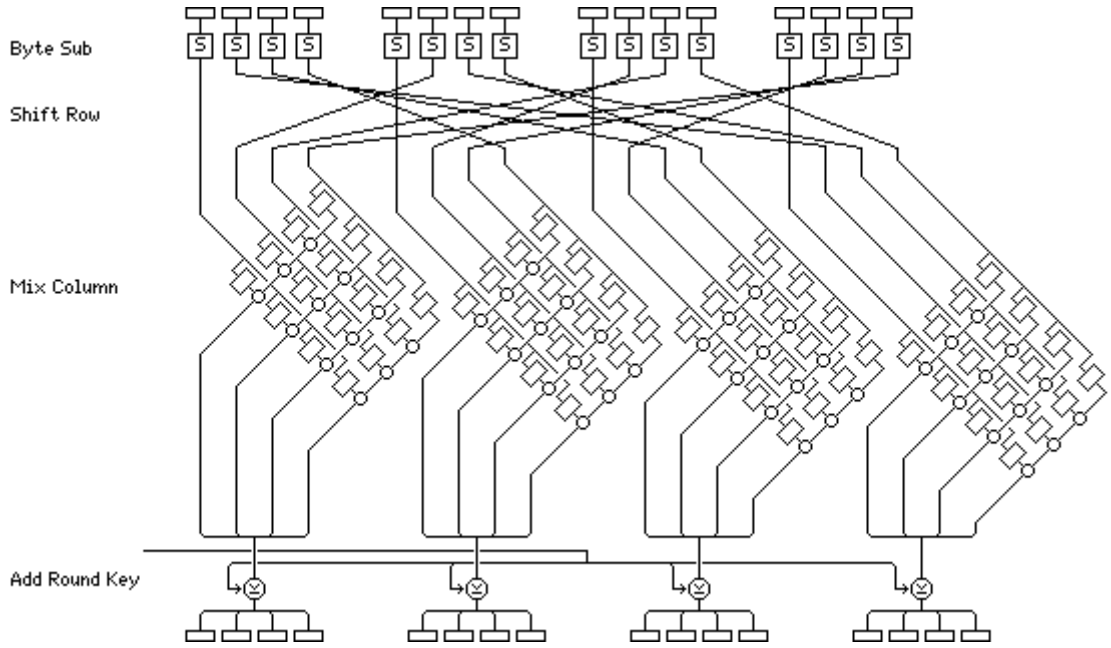


Figura 1. O iterație a algoritmului Rijndael.

Din ultima iterație este omis pasul de amestecare a coloanelor.

2.2. Decriptarea

Pentru a decripta mesajul fabricat de algoritmul Rijndael este necesar ca operațiile descrise să fie înlocuite cu operațiile lor inverse şi ca acestea să fie aplicate în ordine inversă (prima operație din algoritmul de decriptare trebuie să fie inversa ultimei operații din algoritmul de criptare). Succesiunea paşilor în algoritmul Rijndael este:

ARK BS SR MC ARK BS SR MC ARK ... BS SR MC ARK BS SR ARK

Deşi această secvență nu este simetrică, ordinea unor operații poate fi modificată fără ca procesul de criptare să fie afectat. De exemplu pasul de substituire a octeților BS (notat cu B în continuare), poate fi la fel de bine făcut şi după pasul de deplasare a liniilor SR (notat cu S în continuare). Această observație este utilă pentru procesul de decriptare. Făcând această inversare secvența algoritmului, de forma:

A BSMA BSMA ... BSMA BSA

se transformă într-o secvență de forma:

A SBMA SBMA ... SBMA SBA (1R)

Pentru fiecare pas s-a folosit notația bazată pe prima literă a denumirii engleze a pasului. Dacă se inversează secvența care descrie algoritmul se obține:

ASB AMSB ... AMSB AMSB A (2R)

Comparând secvențele (1R) şi (2R) se constată că pe lângă diferita poziționare a spațiilor (acestea marchează începutul unei noi iterații a algoritmului de criptare) singura diferență care mai apare este că grupurile "MA" din (1R) sunt înlocuite cu grupuri "AM" în (2R).

E clar că nu este suficientă inversarea ordinii paşilor folosiți la criptare pentru a se face decriptarea ci trebuie inversate şi operațiile care compun aceşti paşi. Pentru inversarea pasului ARK trebuie inversată funcția sau-exclusiv. Dar această inversare se realizează tot cu funcția sau-exclusiv. De aceea pasul ARK nu trebuie inversat la decriptare. Nu acelaşi lucru se poate spune despre ceilalți paşi. Este de exemplu cazul pasului de amestecare a coloanelor, MC, (notat cu M în relațiile (1R) şi (2R)) pentru inversarea căruia, în procesul de decriptare este necesară inversarea matricii cu care se înmulțeşte fiecare vector. La fel trebuie procedat şi cu matricea cutiei de tip S din pasul de substituție a octeților, BS (notat cu B în relațiile (1R) şi (2R)).

Revenind la relațiile (1R) şi (2R) este legitimă întrebarea: Trebuie inversată ordinea secvenței paşilor "MA" şi "AM" pentru decriptare ?
Răspunsul este Nu, deoarece operația de înmulțire a matricilor este distributivă în raport cu operația de adunare pe câmpul Galois al lui 2^8 . Operația de sau-exclusiv din cadrul pasului MC (M) este de fapt identică cu operația de adunare definită pe câmpul Galois al lui 2^8 . De aceea cheile de iterație, implicate în procesul de inversare al pasului de amestecare a coloanelor, trebuiesc înmulțite cu inversa matricii de amestecare a coloanelor şi apoi se pot calcula funcțiile sau-exclusiv, la fel ca la criptare (bineînțeles cheile de iterație trebuiesc luate în ordine inversă în raport cu ordinea folosită la criptare). Matricea pentru inversarea pasului de amestec al coloanelor este:

14 11 13 9
9 14 11 13
13 9 14 11
11 13 9 14

iar forma sa binară, folosită în algoritmul de decodare este:

1110 1011 1101 1001 01 00 00 00
1001 1110 1011 1101 00 01 00 00
1101 1001 1110 1011 00 00 01 00
1011 1101 1001 1110 00 00 00 01

111 101 110 100 01 01 00 00
110 100 111 101 00 00 01 01
1100 1000 1110 1010 00 00 10 10
1011 1101 1000 1110 01 01 10 10
0 0 1 0 01 01 10 11

2.3. Generarea cheilor

Pentru cazul în care se foloseşte o cheie inițială cu lungimea de 128 biți sau de 192 de biți, toate subcheile necesare pentru toate iterațiile, se obțin din cheia inițială (prima subcheie fiind chiar cheia inițială) sau din variante ale cheii inițiale şi au aceeaşi lungime cu aceasta.

Subcheile sunt alcătuite din cuvinte de 4 octeți. Fiecare cuvânt se obține calculând sau-exclusiv între cuvântul anterior de 4 octeți şi cuvântul corespunzător dintr-o variantă anterioară sau rezultatul aplicării unei funcții acestui cuvânt (din varianta precedentă). Pentru stabilirea primului cuvânt dintr-o anumită variantă, cuvântul inițial (cel curent pentru iterația respectivă) este pentru început rotit cu opt poziții spre stânga, apoi octeții săi sunt modificați folosind cutia de tip S din pasul de substituție a biților BS (B) corespunzător, iar apoi se calculează sau-exclusiv între primul octet al rezultatului obținut anterior şi o constantă dependentă de iterație. Constantele dependente de iterație sunt:

```
  1   2   4   8  16  32  64 128
 27  54 108 216 171  77 154  47
 94 188  99 198 151  53 106 212
179 125 250 239 197 145  57 114
228 211 189  97...
```

sau în binar:

```
00000001 00000010 00000100 00001000 00010000 00100000 01000000 10000000
00011011 00110110 01101100 11011000 10101011 01001101 10011010 00101111
01011110 10111100 01100011 11000110 10010111 00110101 01101010 11010100
10110011 01111101 11111010 11101111 11000101 10010001 00111001 01110010
11100100 11010011 10111101 01100001...
```

puterile lui 2 succesive în reprezentarea din câmpul Galois al lui 2^8 folosit.

3. Programul AES

Lucrarea de față se bazează pe utilizarea programului aes.exe. Când se face *click* pe aes.exe, pe ecranul
monitorului apare fereastra din figura următoare.

Figura 2. Fereastra principală a programului AES.

Făcând *click* pe Teorie, se obține fereastra reprezentată în figura următoare.

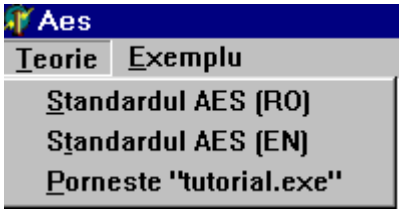
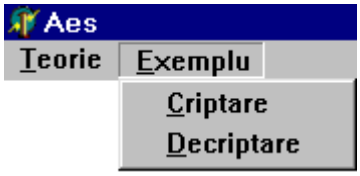


Figura 3. Fereastra Teorie a programului AES.

După cum se poate vedea din figura anterioară, poate fi consultată o prezentare în limba română a algoritmului Rijndael, Standardul AES (RO) şi o prezentare a aceluiaşi standard în limba engleză, Atandardul AES (EN). De asemenea poate fi utilizat un program de învățare a algoritmului, tutorial.exe.

Făcând *click* pe Exemplu, se obține fereastra din figura următoare



.

Figura 4. Fereastra Exemplu a programului AES.

Făcând *click* pe Criptare, se obține fereastra din figura următoare.

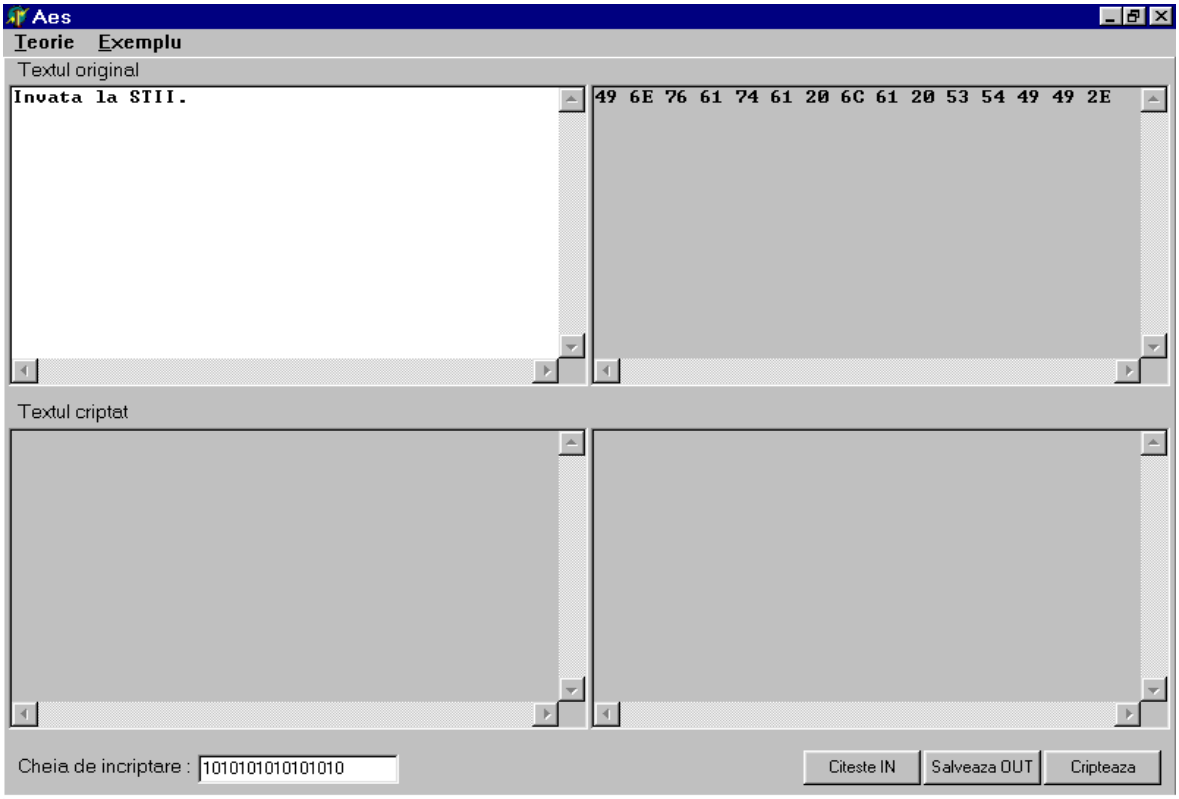


Figura 5. Fereastra de criptare.

Se completează, în fereastra din stânga sus, textul de criptat şi apoi cheia de criptare. În fereastra din dreapta

sus apare textul de criptat, codat în hexazecimal. Apoi se apasă pe butonul "Criptează", obținându-se fereastra din figura următoare.

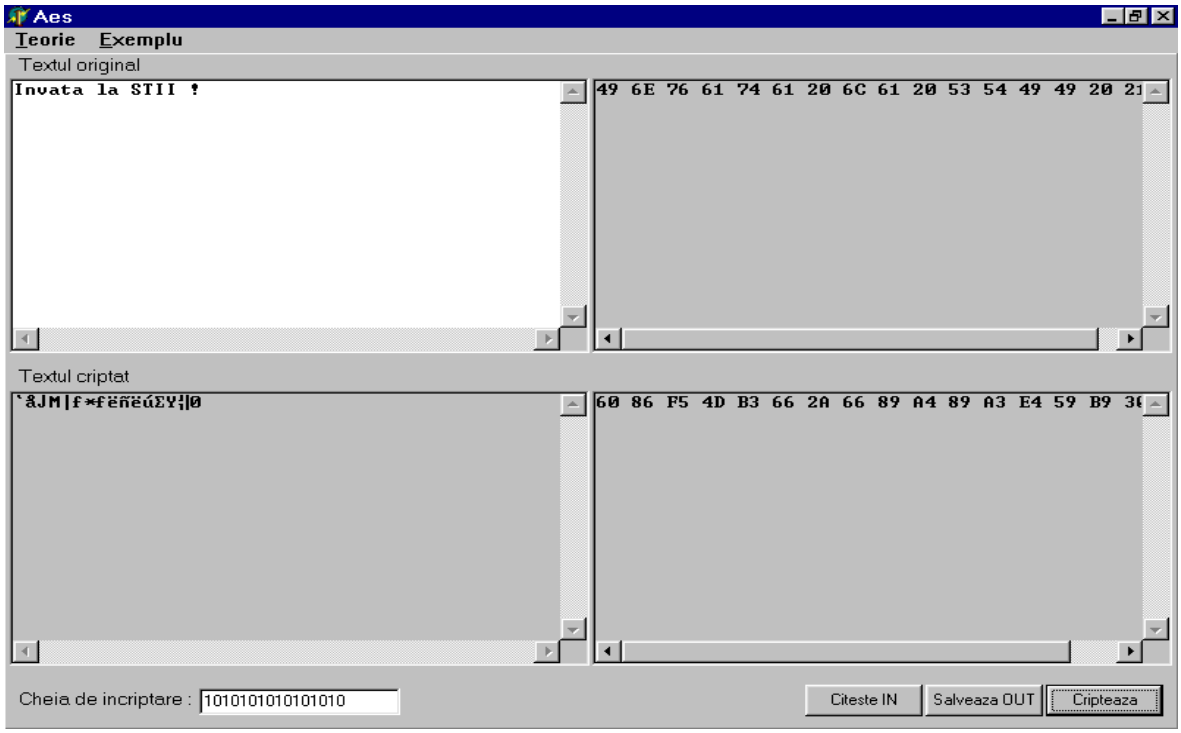


Figura 6. Rezultatul criptării.

Textul criptat este prezentat în fereastra din stânga jos. În fereastra din dreapta jos este prezentat textul criptat codat în hexazecimal.

Pot fi codate şi fişiere. În acest scop, se apasă pentru început pe butonul "Citeste IN". Cu ajutorul ferestrei care se deschide, în urma acestei acțiuni, se selectează fişierul care se doreşte a fi criptat. Acesta poate fi de tip .txt, .doc, .html, etc. După specificarea cheii de criptare, se apasă butonul "Criptează". Fişierul criptat poate fi salvat. În acest scop trebuie apăsat butonul "Salveaza OUT".

Pentru a decripta un fişier, se alege, în fereastra din figura 4, opțiunea Decriptare. Apoi se poate selecta, cu ajutorul opțiunii "Citeste IN", fişierul care se doreşte a fi decriptat. După această selecție, se specifică cheia de decriptare (care trebuie să fie identică cu cea care a fost folosită pentru criptare). În fereastra din stânga jos apare textul criptat, iar în fereastra din dreapta jos, textul criptat, codat în hexagesimal. Apoi se apasă butonul "Decripteaza". În fereastra din stânga sus apare textul decriptat, iar în fereastra din dreapta sus, varianta codată în hexagesimal a acestuia. Bineînțeles că şi acest text poate fi salvat dacă se apasă butonul "Save OUT".

4. Desfăşurarea lucrării

4.1.Se verifică toate funcțiile programului evidențiate în paragraful anterior. În acest mod se învață să se lucreze cu programul.

4.2.Se parcurge programul tutorial.exe .

4.3. Se efectuează verificarea funcționării programului efectuând o criptare şi o decripate. După efectuarea criptării se notează textul în clar (în cele două forme ale sale: şir de caractere şi hexazecimal) şi se salvează şirul de date obținut în urma criptării, într-un fişier text. Acesta se foloseşte la decriptare, urmând ca după aceasta să se regăsească textul în clar folosit la criptare. Mesajul în clar şi criptat vor fi salvate într-un fişier Word, denumit cu numele utilizatorului şi indicele lucrării de laborator, pentru a putea fi identificat ulterior, în directorul Users.

4.4. Pentru acelaşi mesaj se va simula folosirea a două sisteme de criptare AES, prin alegerea, a două chei inițiale diferite. Pentru fiecare dintre aceste experimente se vor nota formele în clar şi criptată ale mesajului. Se vor compara cele două mesaje criptate obținute astfel numărându-se caracterele comune (care se găsesc în acelaşi loc şi au aceaşi valoare). Pe baza acestui criteriu se va stabili care dintre cele două alegeri de parametri a fost mai inspirată. Şi comentariile inspirate de acest experiment vor fi consemnate în fişierul Word cu rezultatele lucrării, amintit mai sus.

Lucrarea 6. Tehnici de balizare utilizând transformarea “wavelet”

1. Scopul lucrării

Balizarea este o tehnică de autentificare a imaginilor. Prin inserarea unei balize invizibile într-o imagine, înainte ca aceasta să fie difuzată şi prin extragerea balizei după recepția acesteia la utilizator, poate fi autentificat dreptul de proprietate auspra imaginii respective al celui care a difuzat-o. În acest mod pot fi identificați şi utilizatorii ilegali ai unei anumite imagini. Pentru realizarea balizării este necesar să se genereze o baliză invizibilă, să se insereze această baliză în imaginea care trebuie difuzată şi să se poată extrage din imaginea recepționată de utilizator. În cazul în care un utilizator ilegal utilizează imaginea respectivă, pentru ca aceasta să nu poată fi autentificată, ar fi necesar ca baliza conținută în aceasta să fie îndepărtată. O balizare de calitate trebuie deci să fie rezistentă la atacurile unor utilizatori ilegali. În lucrarea de față se studiază o metodă de balizare adaptivă (baliza generată este dependentă de imaginea de difuzat).

2. O metodă de balizare

O modalitate de a insera o baliză într-o imagine are la bază utilizarea transformării imaginii. Cea mai des folosită transformare este DCT (transformarea cosinus discretă).

Necesitatea de a face invizibilă baliza face dificil procesul de balizare, rezultând proceduri complicate de prelucrare a imaginii. Din acest motiv, inserarea balizei în domeniul transformatei DCT trebuie să respecte unele condiții perceptule, impuse de regulă sistemului ce realizează cuantizarea în domeniul DCT.

Utilizarea transformării “wavelet” discretă (DWT) în procesul de balizare a imaginilor aduce unele avantaje față de transformarea DCT. Astfel, transformarea DWT a unei imagini este tot o imagine cu aceleaşi dimensiuni cu cele ale imaginii originale, dar care constă din două zone importante:

- zona de aproximare numită şi rezumat, de dimensiuni mai reduse în raport cu imaginea originală;
- zona cu detalii care constă într-un set de imagini de dimensiuni reduse ce conțin detaliile imaginii originale.

Rezultă deci că transformarea DWT oferă acces direct asupra detaliilor unei imagini. Acest lucru permite utilizarea unei proceduri simple şi rapide de inserare a balizei în imagine prin modificarea detaliilor imaginii, păstrând în acelaşi timp transparența perceptuală a balizării. Din tehnicile de balizare ce utilizează transformarea DWT sunt superioare celor ce utilizează transformarea DCT.

Aşa cum s-a arătat anterior, o tehnică simplă de balizare constă în modificarea detaliilor unei imagini, echivalentă cu o modulare în amplitudine a coeficienților transformării DWT corespunzători. În cele ce urmează se va prezenta un mod de implementare în Matlab a acestei metode de balizare, beneficiind de suportul oferit de pachetul Wavelab în domeniul transformării DWT.

3.1. Algoritmul de inserare a balizei in imagine

Balizarea imaginii se realizează conform schemei bloc din figura 1.

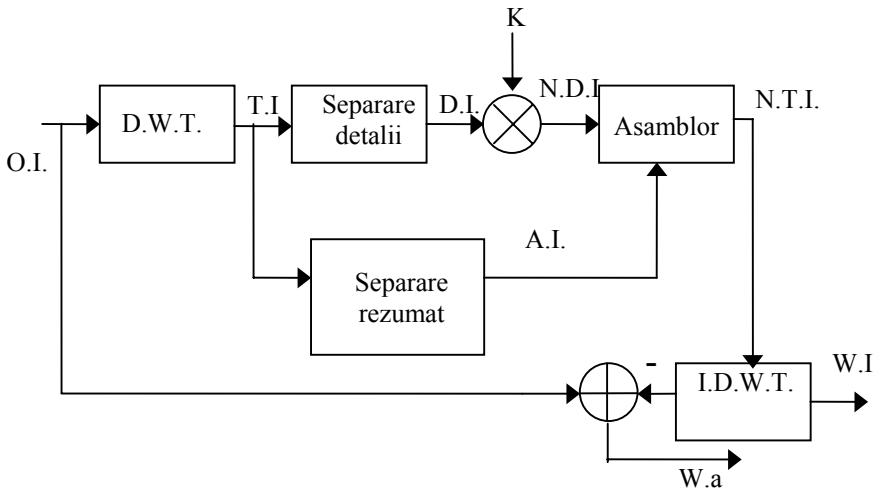


Figura 1. Schema de balizare.

şi constă din următoarele etape:

- calculul transformatei DWT a imaginii originale O.I., T. I.;
- separarea detaliilor şi a rezumatului din cadrul T.I. (D.I. şi respectiv A.I.);
- multiplicarea detaliilor cu constanta K (N.D.I.);
- asamblarea imaginii balizate în domeniul transformatei DWT, din rezumat şi din detaliile multiplicate cu K (N.T.I.);
- calculul transformării DWT inverse în vederea obţinerii imaginii balizate (W.I);
- obţinerea balizei prin calculul diferenţei dintre imaginea originală şi imaginea balizată (W.a).

În continuare se prezintă succint modul în care are loc separarea rezumatului de detalii pentru o imagine dată, precum şi reasamblarea lor după inserarea balizei. Aşa cum s-a arătat anterior transformarea DWT a unei imagini este compusă din două zone principale, ca în figura 2.

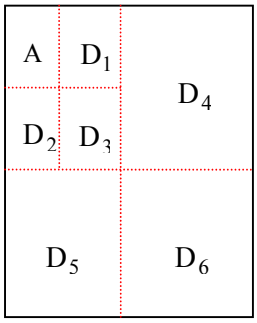


Figura 2. Transformarea DWT a unei imagini.

Zona delimitată de blocurile A, D_1, D_2 şi D_3 reprezintă rezumatul imaginii rezultate în urma transformării DWT, în timp ce zona delimitată de blocurile D_4, D_5 şi D_6 reprezintă detaliile. Numărul de blocuri ce revine fiecărei zone în parte depinde de numărul de iteraţii

din calculul transformării DWT. Mărimea blocurilor poate fi aleasă după dorință, singura cerință fiind ca ele să nu aparțină simultan celor două zone definite anterior. După multiplicarea cu constanta K (aleasă în așa fel încât să se asigure transparența perceptuală) a coeficienților DWT din blocurile D_4, D_5 şi D_6, se obțin blocurile D_4', D_5' şi D_6' ce conțin deja baliza. Asamblarea blocurilor noi obținute se face ca în figura 3, pentru a putea obține în urma transformării DWT inverse imaginea balizată.

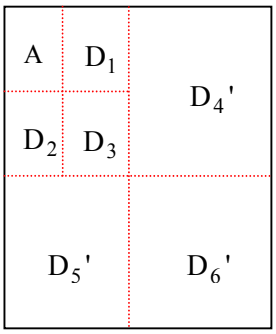


Figura 3. Asamblarea blocurilor de imagine după inserarea balizei.

După cum s-a putut observa, această metodă de balizare este adaptivă, deoarece depinde de conținutul imaginii originale (sursă). În ce priveşte valoarea constantei K, este relativ uşor de determinat valoarea ei în așa fel încât balizarea să fie imperceptibilă. Prin urmare nu este necesară utilizarea de tehnici suplimentare pentru a asigura transparența perceptuală.

3.2. Algoritmul de extragere a balizei

Extragerea balizei dintr-o imagine balizată utilizând algoritmul prezentat în paragraful 3.1. se face cu schema din figura 4.

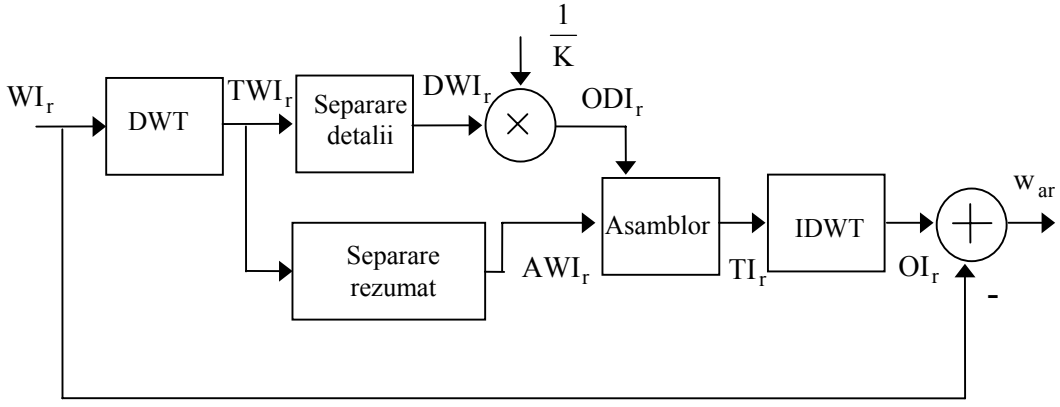


Figura 4. Schema de extragere a balizei

Paşii parcurşi pentru extragerea balizei sunt similari cu cei de la balizare:

- calculul transformatei DWT a imaginii balizate;
- separarea zonelor cu rezumat şi respectiv cu detalii ale imaginii;
- înmulțirea detaliilor cu constanta 1/K;

- reasamblarea zonelor cu rezumat şi a celor cu detalii rezultate după multiplicarea cu K;
- calculul transformatei DWT inverse pentru obţinerea imaginii originale.
- calculul balizei ca diferenţă dintre imaginea balizată recepţionată şi cea originală obţinută în urma extragerii balizei.

În cazul în care imaginea balizată utilizată de algoritmul de extracţie este identică cu cea obţinută la balizare, balizele obţinute în procesul de inserare şi extracţie sunt identice. Dacă apar erori de transmisie a imaginii balizate, sau prelucrări/atacuri asupra imaginii balizate, baliza extrasă nu va mai fi identică cu baliza obţinută în cadrul procesului de inserare a balizei. Dacă algoritmul de balizare este robust, diferenţa dintre cele două balize trebuie să fie mică. Pentru a caracteriza gradul de asemănare a celor două balize în vederea identificării, se defineşte factorul de asemănare ca fiind factorul de corelaţie, cu relaţia:

$$f_c = \frac{\sum_m \sum_n w_a[m,n] \cdot w_{ar}[m,n]}{\sqrt{\sum_m \sum_n w_a^2[m,n] \cdot \sum_m \sum_n w_{ar}^2[m,n]}}$$

Valoarea factorului de corelaţie este unitară atunci când balizele de la inserare şi extracţie sunt identice, şi scade spre zero atunci când apar diferenţe. Ea serveşte ca măsură a robusteţii algoritmului de balizare la prelucrări şi atacuri asupra imaginii balizate. Totodată, valoarea sa poate fi folosită ca şi criteriu de decizie pentru a stabili dacă în imaginea analizată se află baliza căutată. Pentru aceasta este nevoie să se stabilească o valoare de prag (de ex. 0.7) peste care se decide că baliza extrasă este cea căutată, în caz contrar neputându-se face identificarea certă.

4. Desfăşurarea lucrării

1.

Din Windows Commander se selectează directorul compwater.m. Se citeşte cu F4. Se selectează textul (Edit, Select All) şi se copiază (Edit, Copy). Se deschide MATLAB-ul. Se copiază textul selectat anterior în fereastra de lucru a MATLAB-ului (Edit, Paste). Se rulează acest program (Enter). Se salvează în directorul USERS (personal) rezultatele obţinute (cele 4 imagini: imaginea originală, imaginea transmisă, baliza generată la emisie şi baliza generată la recepţie).

2.

Se studiază programul Matlab utilizat, citind (cu F4) fişierul compwater.m şi identificând principalele etape ale algoritmilor de inserare, respectiv extragere a balizei.
Se vor comenta rezultatele obţinute.

3.

Se repetă punctele anterioare pentru o altă valoare a lui k, de exemplu 2. În acest scop se modifică linia 13 a programului compwater.m.

4.

Se repetă punctele anterioare pentru o altă imagine, de exemplu: Lenna. În acest scop se modifică linia a doua a programului compwater.m, aceasta devenind:
ingrid=readimage('Lenna').

Lucrarea 7. Protecţia poştei electronice folosind pachetul de programe PGP, *Pretty Good Privacy*

1. Scopul lucrării

Se studiază modul în care pot fi criptate mesajele electronice folosind programul PGP.

2. Pachetul de programe PGP, *Pretty Good Privacy*

Acest pachet a fost conceput de Phil Zimmermann. Deoarece acesta a fost suspectat că ar fi încălcat interdicţia impusă de guvernul american asupra exportului de produse criptografice, el a fost urmărit în justiţie timp de mai mulţi ani. În prezent este patronul unei companii de *software* care comercializează acest pachet de programe.

Funcţionarea PGP

PGP combină câţiva dintre cei mai buni parametri ai criptografiei simetrice şi asimetrice. El este un sistem de criptare hibrid. Când un utilizator criptează un text în clar cu PGP, acesta comprimă prima dată textul în clar. Compresia creşte rezistenţa la atacuri de criptanaliză. Apoi PGP crează o cheie de sesiune care este folosită o singură dată. Această cheie este un număr aleator. Ea lucrează în acord cu un algoritm de criptare foarte sigur şi rapid pentru a cripta varianta comprimată a textului în clar. Rezultatul este textul criptat. De îndată ce datele au fost criptate, este criptată şi cheia sesiunii, folosindu-se cheia publică a destinatarului. Varianta criptată a cheii sesiunii este transmisă împreună cu textul criptat. Pentru decriptare se aplică operaţiile dscrise în ordine inversă. Destinatarul recepţionează mesajul PGP, îşi foloseşte cheia secretă pentru a reconstrui cheia sesiunii, pe care apoi programele PGP o folosesc pentru a decripta textul criptat.

Chei

PGP memorează cheile în două fişiere de pe *hard disk*-ul calculatorului gazdă. Unul dintre ele este folosit pentru cheile secrete iar celălalt pentru cheile publice. Acestea se numesc inele de chei *keyrings*. Dacă un utilizator îşi pierde cheia secretă, el nu va mai putea să decripteze nici un mesaj PGP pe care îl primeşte.

Semnături digitale

Se folosesc pentru autentificarea sursei mesajului şi pentru verificarea integrităţii acestuia. Ele asigură şi nerepudierea mesajului. În figura următoare este exemplificat modul de generare a unei semnături digitale.
În loc să se cripteze informaţia cu cheia publică a cuiva se foloseşte cheia secretă a utilizatorului. Dacă acea informaţie poate fi decriptată cu cheia publică a utilizatorului atunci înseamnă că a fost generată de către acesta.

PGP foloseşte funcţia *hash* MD-5 pentru a obţine un rezumat (*message digest*) al textului în clar pe care trebuie să-l semneze utilizatorul. Cu ajutorul acestui rezumat şi al cheii secrete a utilizatorului, acesta crează semnătura. PGP transmite împreună semnătura şi mesajul în clar. După recepţie destinatarul foloseşte PGP pentru a recompune rezumatul, verificând în acest fel semnătura. Mesajul în clar poate fi criptat sau nu. Semnarea unui text în clar este utilă dacă unii dintre destinatari nu sunt interesaţi sau nu sunt capabili să verifice semnătura.

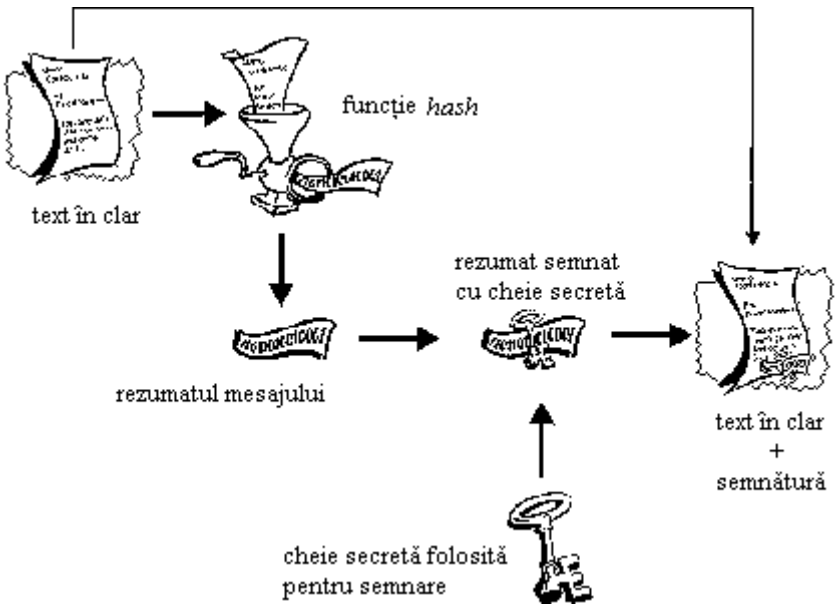


Figura 1. Generarea unei semnături digitale.

Atât timp cât se utilizează o funcție *hash* sigură, nu există nici o posibilitate să se copieze semnătura cuiva dintr-un mesaj şi să se ataşeze într-un altul sau se altereze un mesaj semnat. Cea mai mică modificare a unui document semnat va cauza insuccesul procesului de verificare a semnăturii. Semnăturile digitale joacă un rol important în autentificarea şi validarea cheilor unor noi utilizatori PGP. În figura următoare se prezintă procesul de generare şi inserare a unei semnături digitale.

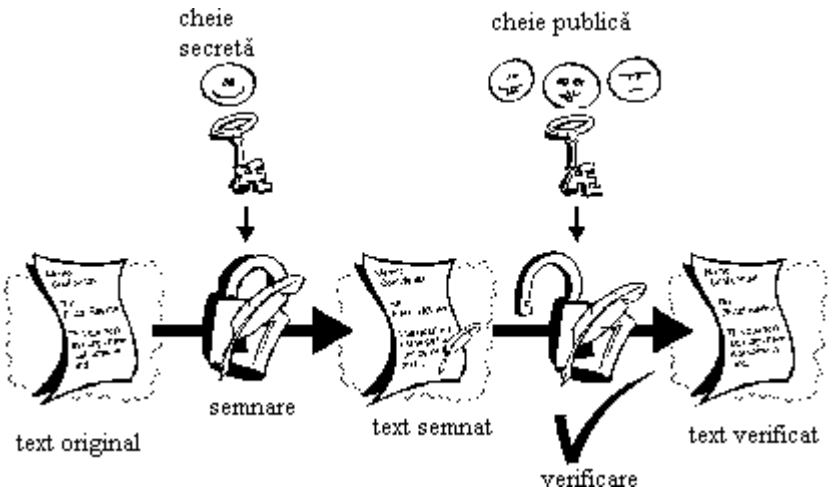


Figura 2. Procesul de inserare a semnăturii digitale.

Ce este o parolă de tip frază ?

O parolă de tip frază este o variantă mai lungă de parolă care este folosită de către un utilizator în scop de identificare. Aceasta este mai sigură împotriva atacurilor bazate pe forța brută. PGP foloseşte o parolă de tip frază pentru criptarea cheii unui utilizator pe propriul calculator. Nu este permis ca utilizatorul să-şi uite parola de tip frază.

Împărtăşirea cheilor

Se spune că un secret nu mai este secret dacă este cunoscut de două persoane. La fel este şi în cazul unei chei secrete. Deşi nu este recomandabil uneori este necesar să se utilizeze în comun chei secrete. În aceste situaţii este recomandabil ca porţiuni ale cheii secrete să fie făcute cunoscute câte unei persoane, astfel încât acea cheie să poată fi folosită doar cu participarea tuturor acelor persoane.

3. Desfăşurarea lucrării

1. Se instalează programul de poştă electronică, Eudora, făcând *click* pe Setup.exe, din directorul Disk1us. Se configurează programul instalat. Se instalează programul PGP, făcând *click* pe PGPDesktop710Eval30.exe1. Se configurează programul PGP. În acest scop se citeşte fişierul ReadMe din directorul c:\ Program Files \ Network Associates \ PGP for Windows ** \ . Apoi se deschide programul Eudora şi se face *click* pe PGP. Se citesc Help Topics şi apoi se configureaza (respectiv se verifica configurarea), dupa ce s-a apasat pe butonul Options. O parte a unui exemplu de configurare este prezentată în figurile următoare:

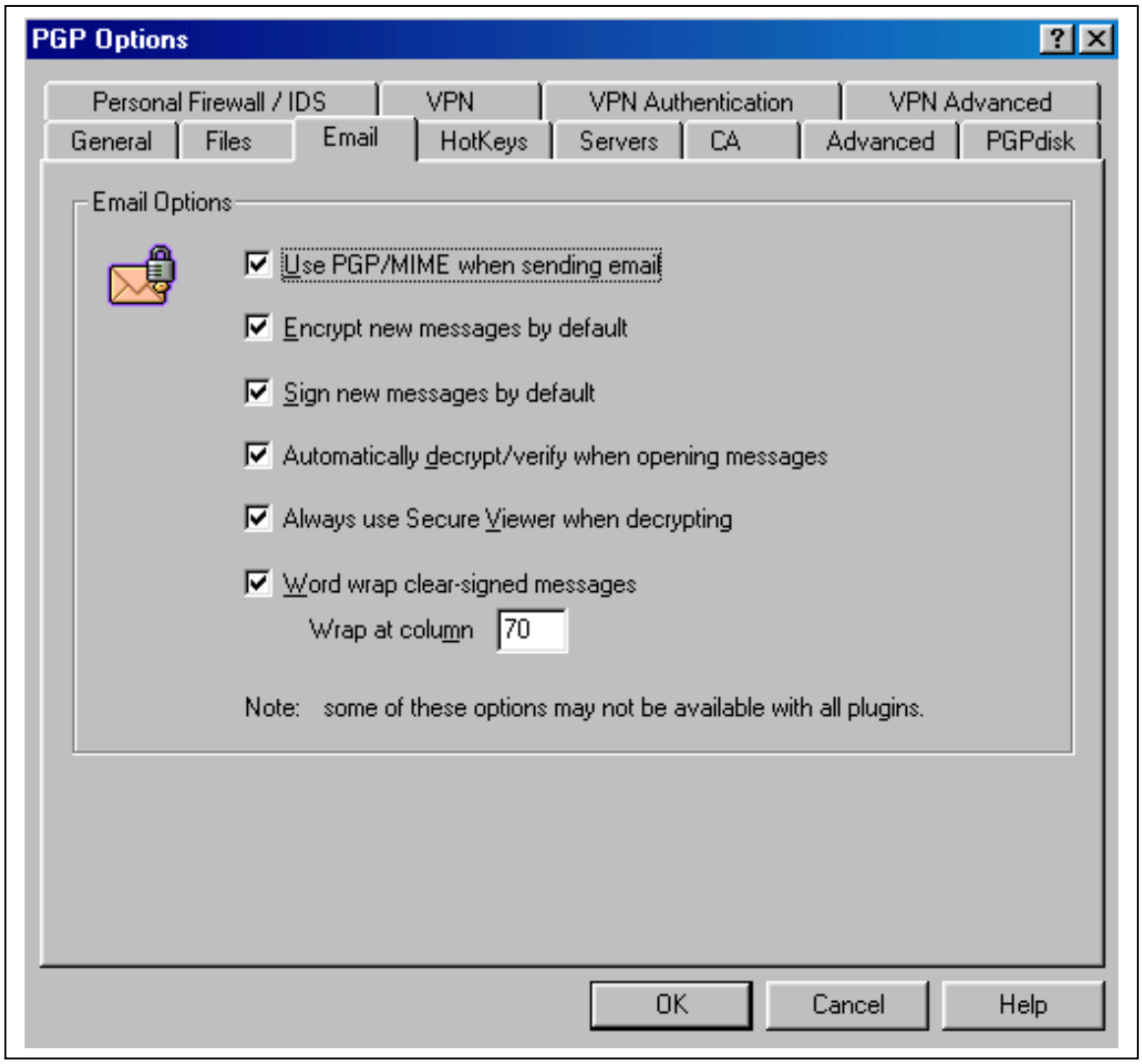


Figura 3. Configurarea programului de poştă electronică.

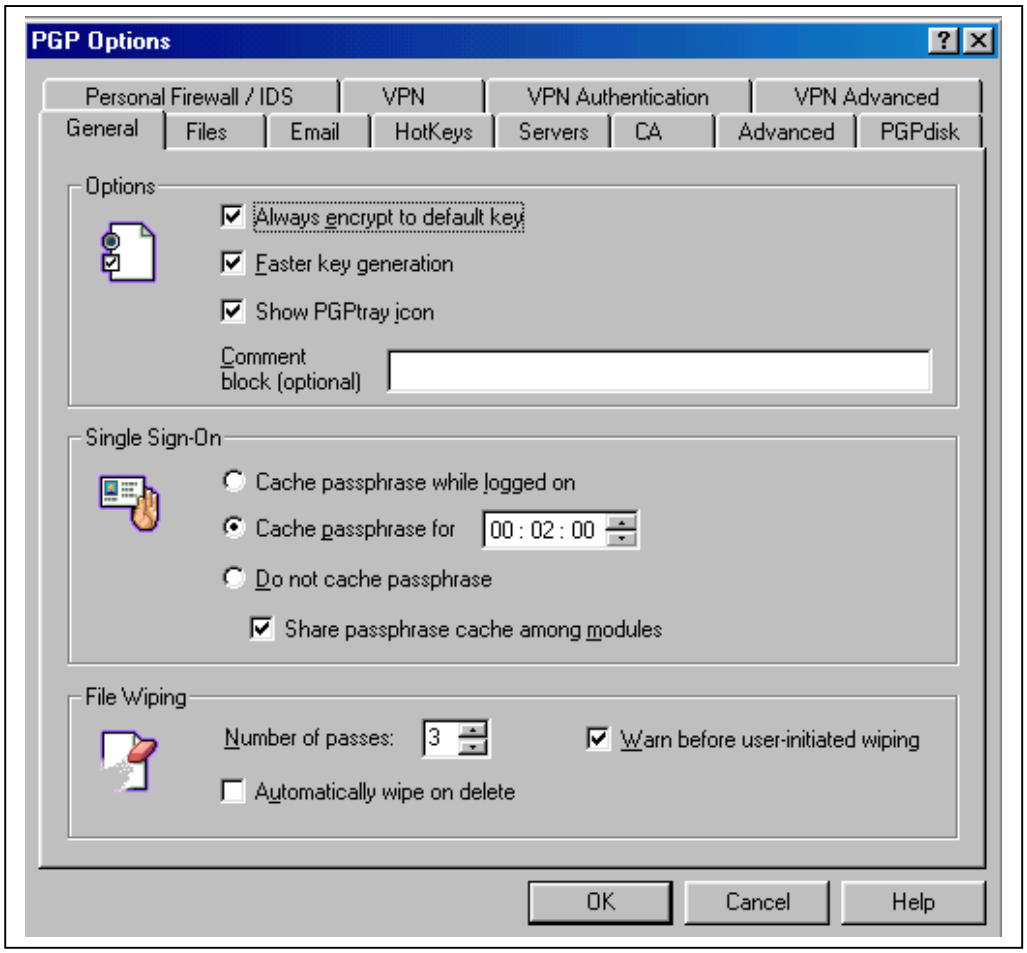


Figura 4. Configurare generală.

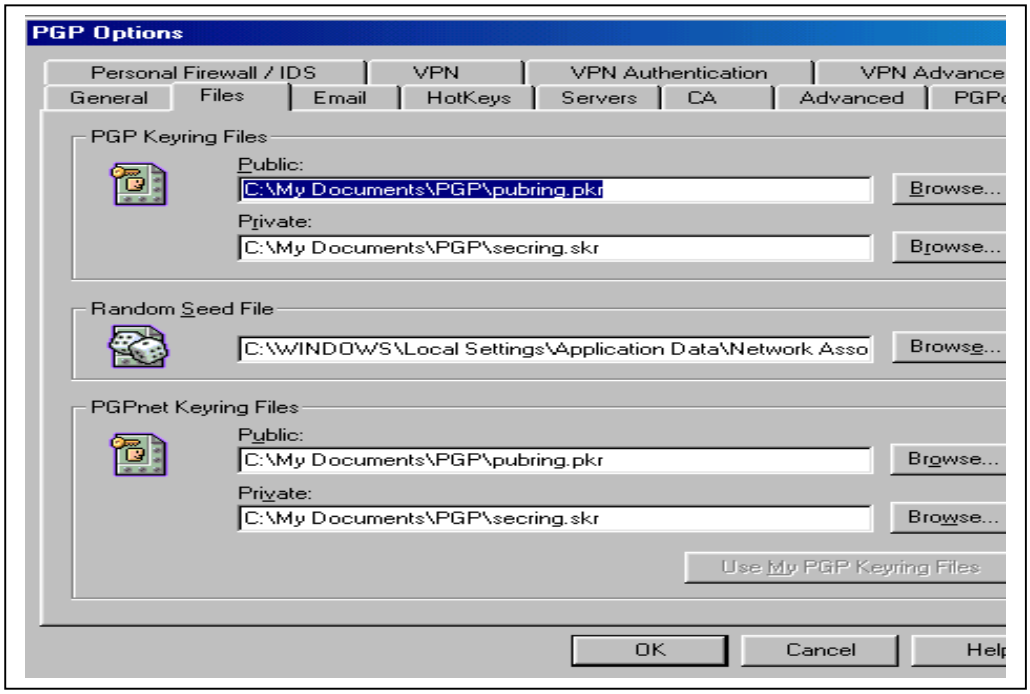


Figura 5. Configurarea fişierelor.

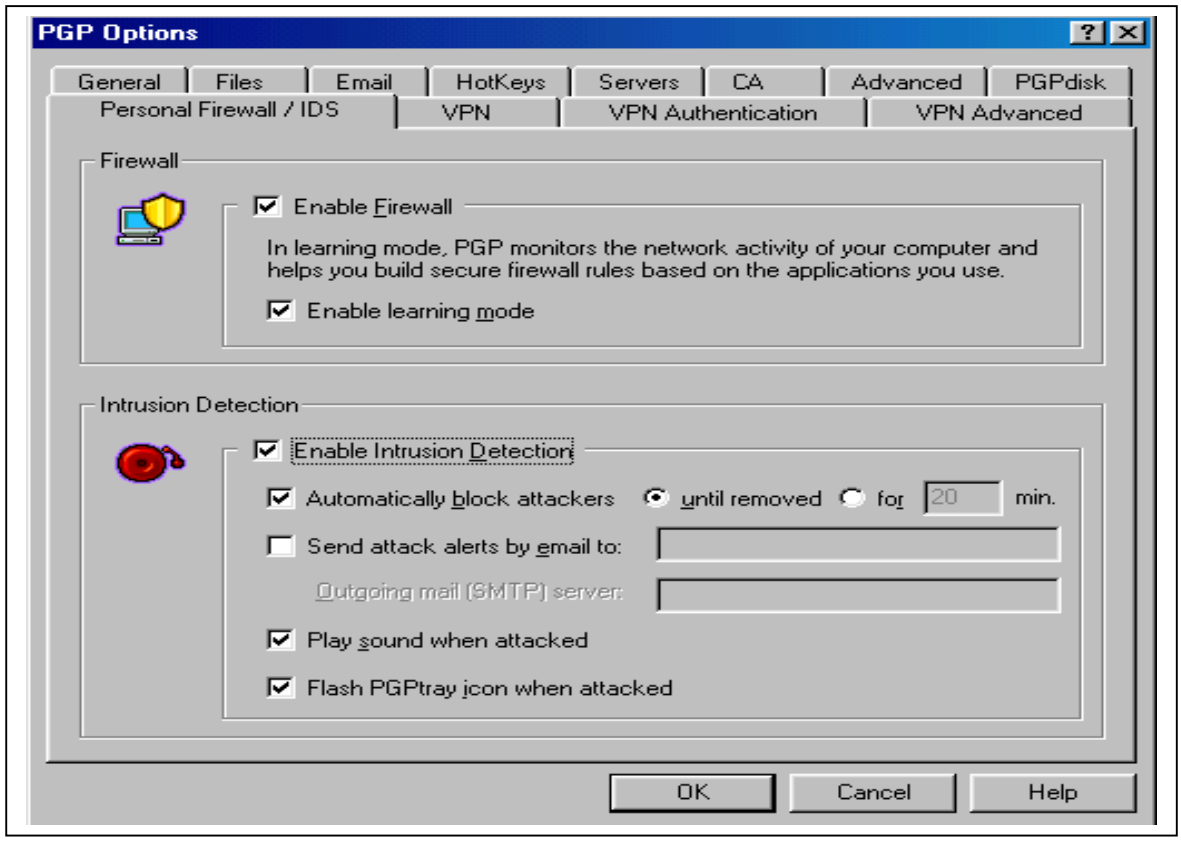


Figura 6. Configurarea firewall-ului.

2. După configurare se deschide Eudora se selectează *Message* şi apoi *New Message.* Se selectează *PGP keys* şi se apasă pe *Server* şi apoi pe *Send to,* alegându-se pgpkeys.mit.edu.

3. Se transmite un mesaj criptat unui utilizator care nu are instalat programul PGP. În acest scop se deschide Eudora se selectează *Message* şi apoi *New Message.* Se scrie mesajul se selectează *PGP Encrypt Email Message* (celelealte opțiuni se vor deselecta) şi apoi *Send.* Se constată că forma criptată a mesajului nu poate fi înțeleasă de către acesta.
4. Se transmite acelaşi mesaj unui utilizator care are instalat programul PGP. Se verifică faptul că acesta poate înțelege mesajul.
5. Se repetă cele două operații, descrise anterior pentru o nouă alegere a parolei de tip frază.
6. Se dezinstalează programele Eudora şi PGP.