Capitolul 8. PREZENTAREA ŞI PROTECŢIA DATELOR

8.1. Rolul nivelului prezentare

Nivelul prezentare include funcţiile legate de reprezentarea datelor transmise, conversia, cifrarea (criptarea) şi compresia, realizate astfel încît să se păstreze semnificaţiile informaţiei transportate de nivelele inferioare.

Conversia datelor este impusă de utilizarea unor reprezentări interne distincte în maşini diferite, interconectate. Regulile de conversie sînt complexe, ele referindu-se nu numai la valori simple (întregi, reale, caractere etc) ci şi la valori structurate, comunicate între aplicaţii din maşini diferite. Ideea care stă la baza realizării acestei funcţii este utilizarea unui format standard al reţelei, denumit **sintaxă de transfer** şi conversia datelor la şi de la acest format, la ambele capete ale oricărei conexiuni. Cheia întregii probleme de reprezentare, codificare, transmitere şi decodificare a structurilor de date este disponibilitatea unui mijloc de descriere a structurilor de date suficient de flexibil pentru a satisface o gamă largă de aplicaţii şi suficient de precis în ceea ce priveşte semnificaţia. O astfel de notaţie, denumită **sintaxă abstractă** ASN.1 (abstract syntax notation 1) a fost elaborată de ISO şi este descrisă de standardul 8824. Standardul asociat 8825 descrie regulile de codificare a structurilor de date ASN.1 într-un şir de biţi pentru transmisie la distanţă (trecerea la sintaxa de transfer).

Compresia datelor este cerută de micşorarea costului transmisiei, care este proporţional cu volumul traficului de date. Ea este utilizată şi din diferite alte motive, cum este economisirea spaţiului de memorie pe disc sau bandă magnetică. Compresia este realizată prin adaptarea lungimii unităţilor de date la valorile conţinute. De exemplu, dacă un procent ridicat de valori de tip întreg sînt cuprinse între 0 şi 250, se va adopta reprezentarea acestora pe un octet şi doar a valorilor ce ies din domeniu, pe o lungime mai mare (de exemplu pe 32 de biţi precedaţi de un octet special cu valoarea 255). Similar, codurile Huffman asociază simbolurilor transmise coduri de lungimi diferite, dependente de frecvenţa apariţiei lor în textele transmise.

Cifrarea (criptarea) este folosită ca mijloc de protecţie împotriva accesului neautorizat la date, pentru a verifica transmiţătorii mesajelor şi pentru semnatură digitală. Criptarea se poate face la orice nivel, dar în practică locurile cele mai potrivite sînt nivelele fizic, transport şi prezentare.

Cea mai mare parte a primitivelor de serviciu ale nivelului pezentare corespund celor ale nivelului sesiune. Trei dintre ele îşi au însă originea în nivelul prezentare, şi anume:

P-U-EXCEPTION-REPORT raportează o excepţie a utilizatorului,

P-P-EXCEPTION-REPORT raportează o excepţie a nivelului prezentare

P-ALTER-CONTEXT schimbă contextul.Ele se referă la gestiunea contextelor, reprezentînd grupuri de structuri de date necesare unei aplicaţii. Nivelul prezentare facilitează negocierea şi schimbarea contextului. La negociere, un utilizator da o listă a bibliotecilor de structuri de date pe care doreşte să le utilizeze, cealaltă parte putîndu-le accepta sau rejecta. In orice moment, oricare parte poate modifica contextul propunînd adăugarea sau scoaterea unor biblioteci din actualul context.

8.2. Reprezentarea şi conversia datelor

8.2.1. Notaţia sintactică abstractă ASN.1

Informaţiile transmise între procesele de aplicaţie nu au totdeauna forma unor şiruri de caractere, ci o varietate de tipuri de date sînt transmise ca unităţi de date ale protocolului de aplicaţie (UDPA). Pe de o parte, cîmpurile unei UDPA au un tip, iar pe de altă parte, în multe cazuri pot fi omise sau au valori implicite. Pentru a codifica datele transmise, toate tipurile de date necesare unei aplicaţii sînt împachetate într-un modul (bibliotecă).

Cînd aplicaţia doreşte să transmită o UDPA ea pasează nivelului prezentare structura de date împreună cu numele ASN.1 al structurii de date. Folosind ca ghid definiţia ASN.1, nivelul prezentare identifică tipul şi dimensiunea diferitelor cîmpuri şi poate face codificarea lor pentru transmisie. La celălalt capăt, entitatea prezentare receptoare găseşte identificatorul ASN.1 al structurii de date (codificat în primul sau primii octeţi) şi poate decodifica diversele cîmpuri aducîndu-le la forma acceptată de calculatorul receptor. Pentru a evidenţia principalele aspecte ale reprezentării şi codificării datelor, să considerăm un tip înregistrare, a cărui definiţie în Pascal este următoarea:

```
type student = record
          nume: array [1..12] of char;
          anul: integer;
          integralist: boolean;
          nascut: integer
     end;
```

Descrierea în ASN.1 pentru acelaşi tip de date este următoarea:

```
student ::= SEQUENCE {
     nume OCTET STRING,--12 characters
     anul INTEGER,
     integralist BOOLEAN,
     nascut INTEGER
}
```

O valoare de acest tip, în reprezentarea internă a unui calculator de 32 de biţi, cu numerotarea biţilor cuvintelor de la dreapta spre stînga (0 = bitul cel mai puţin semnificativ) arată ca în

figura 8.1(a). Aceeaşi valoare, pentru un calculator de 32 de biţi cu numerotarea biţilor unui cuvînt de la stînga la dreapta (bitul 0 cel mai semnificativ) arată ca în figura 8.1(b). Dacă octeţii sînt transmişi în ordinea de la 0 la 23 şi sînt depozitaţi la destinaţie în aceeaşi ordine, obţinem imaginea din

				octet	octet								
E	N	O	I	0	0	I	O	N	E	I	O	N	E
	U	C	S	4	4	S	C	U		S	C	U	
	N	O	I	12	12	I	O	N		I	O	N	
0	0	0	5	16	16	0	0	0	5	5	0	0	0
0	0	0	0	20	20	0	0	0	0	0	0	0	0
0	7	11	2	24	24	0	7	11	2	2	11	7	0

(a) (b) (c)

Figura 8.1

figura 8.1(c), în care valorile cîmpurilor numerice sînt alterate. încercarea de a inversa ordinea în fiecare grup de patru octeţi remediază cîmpurile numerice, dar alterează şirurile de caractere. Se impune deci utilizarea unei notaţii care să conţină tipul şi lungimea fiecărei structuri, pe lîngă valoarea corespunzătoare.

O reprezentare posibilă, utilizînd o codificare a fiecărei structuri prin tip, lungime şi valoare este dată în figura 8.2(a), iar o forma optimizată în figura 8.2(b).

```
5 28 4 12 I O N E S C U _ I O N _ 1 4 0 0 0 5 2 1 0 1 4 0 7 11 2

s  2 s  1                    i 4        b 1   i 4
t  8 i  2                    n          o     n
a    r                       t o        o o   t o
r  o    c                    r c        l c   r c
t  c c  a                    e t        e t   e t
   t a  r                    g e        e e   g e
i  e r  a                      t        a t     t
n  t a  c                      i        n       i
r  i c  t
e    t  e
g    e  r
     r  e
     e
```

(a)

```
5 24 4 11 I O N E S C U _ I O N _ 1 1 5 2 1 0 1 3 7 11 2
```

(b)

Figura 8.2.

După cum s-a menţionat, descrierea ASN.1 a unui tip de date este numită sintaxa abstractă a acestuia, deoarece nu implică

reprezentări specifice. ASN.1 cuprinde tipuri primitive şi constructori pentru realizarea unor tipuri complexe. **Tipurile primitive** sînt următoarele:

INTEGER — întreg de lungime oarecare,
BOOLEAN — TRUE sau FALSE,
BIT STRING — listă de 0 sau mai multi biţi,
OCTET STRING — listă de 0 sau mai multi octeţi,
ANY — uniunea tuturor tipurilor (un cîmp declarat ANY poate căpăta o valoare de orice tip),
NULL — nici un tip; are o singură valoare desemnată tot prin NULL; cînd un cîmp ia această valoare, se consideră că de fapt el nu are nici o valoare, iar la transmiterea înregistrării cîmpul nu trebuie transmis,
OBJECT IDENTIFIER — nume de obiect (de exemplu, o bibliotecă). La stabilirea unei sesiuni, nivelul prezentare gestionează o negociere asupra sintaxelor abstracte, asupra regulilor de codificare şi a protocoalelor ce urmează a fi utilizate de aplicaţie. Toate acestea sînt obiecte (în esenţă biblioteci) şi au asociaţi identificatori, avînd forma unei succesiuni de cuvinte încadrată între paranteze drepte; de exemplu,
[iso standard 8571 part 4 ftam-pci(1)]
desemnează un obiect definit în partea a 4-a a standardului ISO 8571.

Principalii **constructori** utilizaţi în ASN.1 sînt următorii:

SEQUENCE — listă ordonată cu elemente de de diverse tipuri similar unui RECORD în Pascal;
SEQUENCE OF — listă ordonată cu elemente de un singur tip (ca un tablou); nu are limite pentru dimensiuni;
SET — colecţie neordonată de diferite tipuri (ordinea la receptor nu este neapărat aceeaşi cu cea de la transmiţător;
SET OF — colecţie neordonată cu elemente de un singur tip;
CHOICE — orice tip dintr-o listă dată.

ASN.1 are tipuri predefinite; de asemenea există posibilitatea de a declara cîmpuri opţionale (OPTIONAL) sau cu valori implicite (DEFAULT). Existenţa acestora ridică problema identificării datelor la recepţie (care cîmpuri sînt prezente şi care nu sînt?)

ASN.1 permite identificarea oricărei structuri de date sau cîmp, printr-o **etichetă** (tag). Sînt permise patru tipuri de etichete:

UNIVERSAL — rezervat pentru tipurile primitive şi unele tipuri definite;
APPLICATION — utilizat de protocoalele nivelului aplicaţie;
PRIVATE — recomandat pentru tipurile utilizatorilor; specific contextului folosit în interiorul unui anumit tip de date.

Fiecare etichetă constă dintr-un întreg, precedat de unul din cuvintele de tip rezervate, sau de nici un cuvînt, dacă eticheta

este specifică contextului; etichetele sînt incluse în paranteze drepte, ca în exemplul: [APPLICATION 4].

Dacă receptorul poate identifica un element al structurii de date pe baza etichetei, nu mai este necesară transmiterea tipului. Suprimarea transmiterii tipului este indicată prin cuvîntul cheie IMPLICIT, care se pune după etichetă, în descrierea structurii de date. Pentru exemplificare, reluăm structura **student**, în a cărui descriere apar etichete, cîmpuri implicite şi opţionale:

```
  student ::= [PRIVATE 6] IMPLICIT SEQUENCE {
     nume [0] IMPLICIT OCTET STRING,--12 characters
     anul [1] IMPLICIT INTEGER,
     integralist [2] IMPLICIT BOOLEAN DAFAULT TRUE,
     nascut [3] IMPLICIT INTEGER OPTIONAL    }
```

Pe lîngă tipuri, ASN.1 permite definirea valorilor de diferite tipuri. Pentru valoarea folosită în exemplul anterior, avem descrierea:

```
     {"IONESCU ION",5,FALSE,1970}
```

8.2.2. Sintaxa de transfer.

Regula de bază a sintaxei de transfer ASN.1 este utilizarea a patru cîmpuri pentru fiecare valoare transmisă, şi anume:
- un identificator (tip sau etichetă)
- lungimea în octeţi a cîmpului de date
- cîmpul de date
- indicatorul de sfîrşit de date (opţional).

Identificatorul are urm?toarea structură:
- tipul etichetei (2 biţi);
- tip primitiv sau compus (1 bit);
- număr, cuprins între 0 şi 30 (5 biţi); numerele mai mari ca 31 sînt indicate prin valoarea 31 a acestui cîmp şi sînt reprezentate în octetul sau octeţii următori, pe cîte şapte biţi; bitul cel mai semnificativ este 0, cu excepţia ultimului octet în care este 1; tipurile universale au numere predefinite.

Lungimea cîmpului de date are aceeaşi schemă de reprezentare (pe unu sau mai mulţi octeţi) ca numărul identificatorului: lungimi sub 127 de octeţi sînt reprezentate pe un octet, iar cele peste 127 pe mai multi octeţi, folosindu-se pentru reprezentare doar cîte şapte biţi ai fiecărui octet. Dacă datele au lungimea necunoscută, se foloseşte codul 128 în octetul de lungime, asociat cu un indicator de sfîrşit de date.

Codificarea cîmpului de date depinde de tipul acestora. De exemplu, întregii sînt reprezentati în complement faţă de 2, pe un număr de octeţi suficient pentru valoarea acestora. Valorile booleene sînt codificate pe un octet, cu 0 pentru FALSE şi oricare altă valoare pentru TRUE etc.

8.4. Cifrarea datelor

8.4.1. Modelul de bază al criptării

In reţelele de calculatoare cifrarea, sau criptarea datelor este folosită ca mijloc de protecţie împotriva accesului neautorizat la datele transmise prin mediile de comunicare, sau pentru a asigura autentificarea surselor mesajelor. Modelul de bază folosit în cifrarea/descifrarea mesajelor este cel prezentat în figura 8.8.

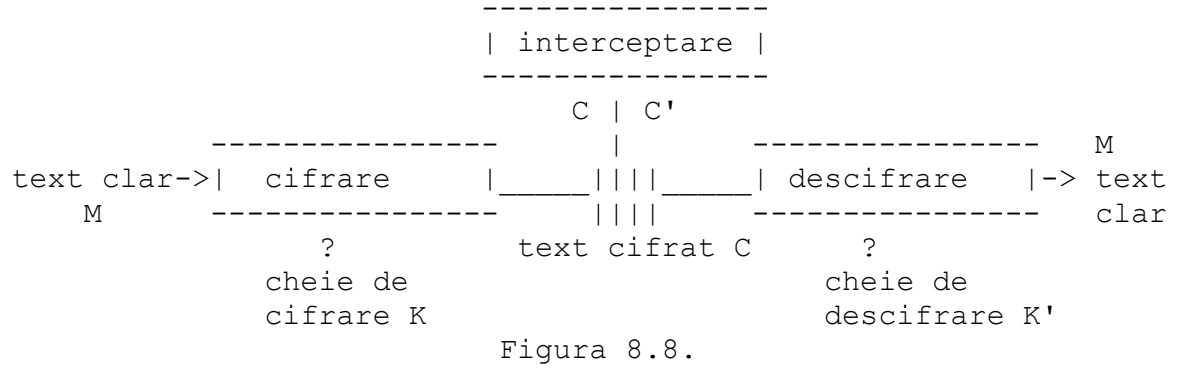


Figura 8.8.

Mesajul necifrat (M), numit şi text clar este supus unei transformări (cifrare sau criptare) care produce un text cifrat sau o criptogramă (C) , folosind o cheie de cifrare (K). Textul cifrat este transmis prin mediul de comunicaţie (nesigur), către destinatarul său. Acesta, realizează transformarea inversă, decifrarea, obţinînd din textul cifrat (C) textul clar (M), folosind o cheie de descifrare (K').

In timpul transmisiei, un intrus poate intercepta textul cifrat. Spre deosbire de receptor, el nu cunoaşte cheia K' de descifrare şi, deci, nu poate descifra textul cu uşurinţă. Dacă reuşeşte să găsească cheia de descifrare K', intrusul poate utiliza în beneficiul propriu informaţia din mesajele interceptate, iar dacă poate determina chiar cheia de cifrare K transmite propriile sale mesaje.

In termenii acestui model, protecţia cere ca intrusul să nu poată reconstitui textul clar dintr-unul cifrat interceptat, deci să nu poată descoperi cheia (secretă) de descifrare K'. Autentificarea cere ca intrusul să nu poată introduce un text cifrat C', fără ca acest lucru să fie detectat, deci să nu poată descoperi cheia (secretă) de cifrare K. Spargerea cifrurilor este subiectul criptanalizei, în timp ce proiectarea lor este domeniul criptografiei. Ambele sînt subdomenii ale criptologiei.

Transformarea realizată la cifrarea unui mesaj poate fi descrisă printr-o funcţie

$$F : \{M\} * \{K\} \rightarrow \{C\}$$

definită pe produsul cartezian al mulţimilor mesajelor şi cheilor, cu valori în mulţimea criptogramelor. Convenţional, se consideră cifrarea ca o transformare E parametrizată după mulţimea cheilor,

astfel că C = Ek(M) iar, descifrarea ca o transformare inversă D, parametrizată după mulţimea cheilor de descifrare, astfel că M = Dk'(C).

Aceasta convenţie are şi o conotaţie de ordin practic. Metoda generală de criptare se consideră stabilă, efortul necesar găsirii şi adoptării unei noi metode fiind foarte mare. Mai mult, de regulă, se consideră ca intrusul (criptanalistul) cunoaşte metoda generală utilizată. Secretul este asigurat prin cheile K şi K' folosite, care pot fi schimbate, teoretic, oricît de des.
Problema criptanalistului are trei variante, care se deosebesc prin informaţiile pe care acesta le deţine:
- criptanaliză cu text cifrat cunoscut, în care se cunoaşte doar un text cifrat;
- criptanaliză cu text clar cunoscut, în care se cunoaşte un text clar şi textul cifrat corespunzător;
- criptanaliză cu text clar ales, în care se cunoaşte modul de cifrare al anumitor porţiuni de text, alese de criptanalist.

8.4.2.Modelul criptografic cu chei publice

Se obişnuieşte să se clasifice sistemele criptografice în **simetrice** şi **asimetrice**. La primele, cheile K şi K' coincid, sau pot fi deduse uşor una din alta. Mai precis, dacă se cunoaşte Ek este foarte uşor de determinat Dk' şi reciproc. Pentru realizarea protecţiei, sau autentificării, trebuie păstrat secretul ambelor chei.

La sistemele asimetrice (propuse de Diffie şi Hellman în 1976) cheile sînt diferite şi nu se pot deduce una din alta. Mai precis, daţi fiind un algoritm de cifrare E şi unul de descifrare D, sînt îndeplinite următoarele condiţii :
- D(E(M)) = M ;
- este extrem de greu să se deducă D din E ;
- E nu poate fi "spart" prin criptanaliză cu text clar ales.

In aceste condiţii, fiecare utilizator U care doreşte să primească mesaje cifrate, face publică cheia (transformarea) Eu de cifrare şi păstrează secretă cheia(transformarea) Du de descifrare. Pentru asigurarea **protecţiei** transmisiei de la un utilizator A la un utilizator B, A trebuie să cifreze orice mesaj M folosind cheia publică Eb a lui B. La recepţie, B face descifrarea folosind cheia sceretă Db (figura 8.9a). Mesajul nu poate fi descifrat de alţi utilizatori, care nu cunosc cheia Db şi nici nu o pot deduce.

Schema nu asigură autentificarea, orice utilizator putînd transmite lui B mesaje, în numele utilizatorului A. Pentru a realiza şi **autentificarea** se poate folosi procedeul schematizat în figura 8.9b. Condiţia necesară aplicării sale este ca transformările Ea şi Da să comute, adică

$$Ea(Da(M)) = Da(Ea(M)) = M.$$

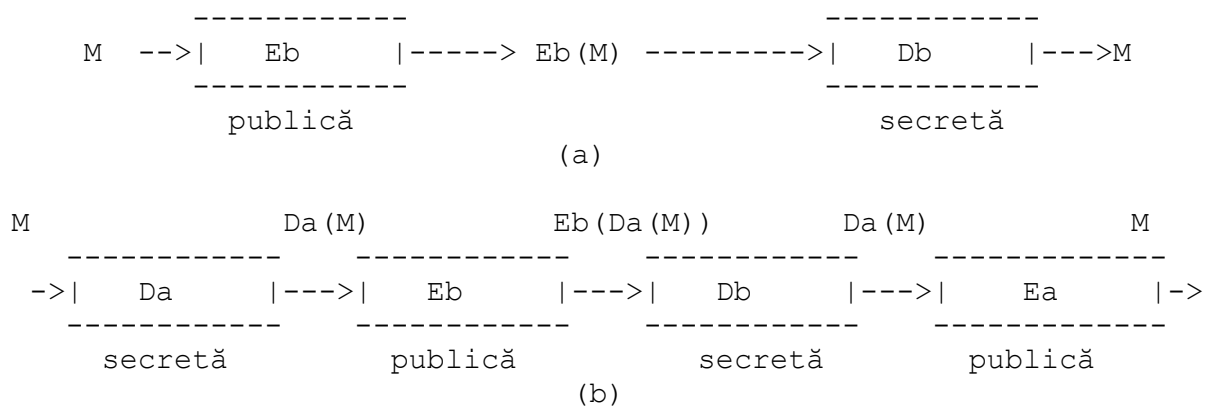


Figura 8.9.

Dacă această condiţie este satisfacută, A transmite lui B un text cifrat obţinut prin aplicarea cheilor Da şi Eb. Cînd B primeşte textul cifrat, el îl transformă folosind cheia sa secretă, obţinînd Da(M), pe care-l păstrează. Apoi descifrează mesajul folosind Ea şi obţinînd textul clar original, M.

Prin această tehnică, B are garanţia că A este sursa mesajului, fiind facută astfel autentificarea. Mai mult, dacă ulterior A contestă că ar fi transmis M, B furnizează unui "judecător" perechea Da(M) şi M. Judecătorul realizează la rîndul său transformarea Ea(Da(M)) şi confruntă rezultatul cu M. Deoarece B nu cunoaşte cheia DA, singura posibilitate ca B să deţină o pereche DA(M) - M corectă este recepţia sa de la A.

După cum se vede, prin transformarea Da(M), utilizatorul A asociază mesajului M o informaţie care permite "identificarea" sa, similară unei semnături de mînă aplicată pe un document. Ea se numeşte **semnătură digitală** şi permite ca :
- receptorul să verifice identitatea transmiţătorului;
-transmiţătorul să nu poată contesta ulterior transmiterea mesajului.

8.4.3.Caracteristicile sistemelor secrete

Pentru ca "atacurile" criptanalitice să fie cît mai dificile, proiectarea cifrurilor trebuie să satisfacă anumite cerinţe, dintre care menţionăm nivelul secretizării şi dimensiunea cheii. Un sistem secret este **necondiţionat sigur** dacă rezistă la orice atac, indiferent de efortul criptanalistului. Un exemplu este banda cu o singură utilizare, la care textul clar este combinat cu o cheie de aceeaşi lungime, aleasă aleator. Datorită lungimii mari a cheii sistemul este nepractic. Din acest motiv, se consideră importante alte caracteristici comportamentale ale sistemelor secrete. Astfel, dacă interceptarea unei părţi oricît de mari din textul cifrat nu simplifică activitatea criptanalistului atunci sistemul este considerat **perfect**. Pe de altă parte, dacă indiferent de volumul textului cifrat interceptat, o criptogramă nu are o rezolvare unică, ci mai multe, cu probabilităţi

apropriate ca mărime, atunci sistemul este **ideal**. Definiţiile prezentate au şi o exprimare formală, care foloseşte noţiuni de teoria informaţiei, aşa cum se arată în lucrarea [16].

Metodele generale de criptare trebuie să satisfacă şi alte cerinţe, care să permită :
-realizarea cifrării şi descifrării prin operaţii cît mai simple;
-reducerea numărului de erori indirecte (produse în lanţ, ca urmare a unei erori de transmisie);- obţinerea unor criptograme de lungimi cît mai reduse, în raport cu lungimile mesajelor clare.

O clasificare riguroasă a metodelor criptografice este greu de făcut, datorită numărului mare de criterii posibile şi a numărului foarte mare de metode propuse sau aflate în uz. O clasificare foarte generală, care ţine cont de evoluţia acestor metode este prezentată în figura 8.10. Principalele caracteristici ale unora din metode sînt prezentate în paragrafele următoare.

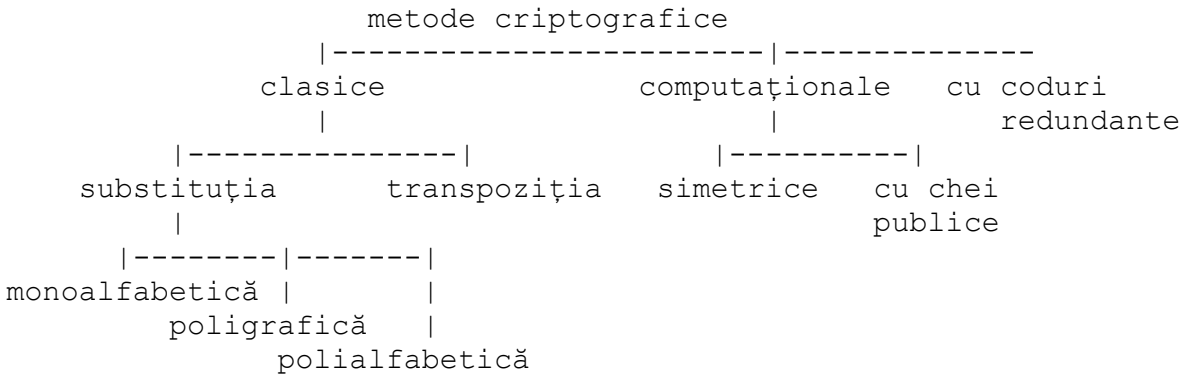


Figura 8.10

8.4.4. Cifrarea prin substituţie

In cifrarea prin substituţie, fiecare literă sau grup de litere din mesaj este înlocuit cu o altă literă sau un alt grup de litere. Un exemplu este **cifrul lui Cezar**, la care alfabetul textului clar coincide cu alfabetul textului cifrat, fiecare literă fiind înlocuită cu cea care îi succede la o distanţă de trei litere, aşa cum se arată în figura 8.11.

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
| | | | | | | | | | | | | | | | | | | | | | | | |
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

Figura 8.11.

Prin această substituţie, textul clar CRIPTOGRAFIE devine FULSWRJUDILH. Din motive lesne de înţeles metoda se numeşte **substituţie monoalfabetică**. Ea are o rezistenţă slabă la atacurile criptanalistului, care pot utiliza pentru dezvăluirea codului

frecvenţele relative de apariţie ale diferitelor simboluri în limbajul natural.

Cifrul lui Cezar poate fi descris prin relaţia

c[i] = (m[i] + 3) mod 26

unde c[i] şi m[i] reprezintă echivalente numerice ale simbolilor alfabetului comun (0 pentru A, 1 pentru B etc) şi poate fi generalizat conform relaţiei

c[i] = (a.m[i] +b) mod n.

Substituţii mai complicate şi deci mai greu de atacat sînt cele poligrafice şi polialfabetice. In substituţia **polialfabetică**, se folosesc prin rotaţie mai multe alfabete de cifrare. Un exemplu este cifrul Vigenere, care foloseşte 36 de cifruri Cezar şi o cheie de cifrare de lungime 1. Cheia, repetată de cîte ori este nevoie, se pune în corespondenţă cu textul clar, literă cu literă. Fiecare literă a cheii determină cifrul Cezar care trebuie folosit pentru litera corespunzătoare din textul clar.

In cifrul Vigenere, cifrul corespunzător unui caracter oarecare X este o permutare circulară a alfabetului mesajelor (A B ....Z), care începe cu X (de exemplu, pentru litera C este C D.....A B).
In exemplul din figura 8.12, cheia utilizată este POLIGRAF.

cheia repetată	POLIGRAFPOLIGRAGPOLIGRAFPOLIGRAFPOLI
text clar	AFOSTODATACANPOVESTIAFOSTCANICIODATA
text cifrat	PTZAZFDFIONITGOATGEQGWOXIQLVOTITSOEI

Figura 8.12

In substituţia **poligrafică**, un grup de n litere este înlocuit cu un alt grup de n litere. Pe măsură ce n creşte, cifrurile se apropie tot mai mult de **coduri**. Spre deosebire de cifruri, care se referă la unităţi de text de lungime fixă, codurile criptează unităţi lingvistice de mărime variabilă (cuvinte sau propoziţii). Corespondenţa între unităţile de text clar şi cele de text cifrat este descrisă în cărţi de cod cu un volum mare şi nu poate fi schimbată la fel de uşor ca o cheie de cifrare.

8.4.5.Cifrarea prin transpoziţie

Cifrarea prin transpoziţie nu înlocuieşte caracterele ci modifică doar ordinea lor. Uzual, textul cifrat se obţine prin dispunerea caracterelor textului clar în liniile succesive ale unei matrice şi parcurgerea acesteia după o anumită regulă pentru stabilirea noii succesiuni de caractere. Un exemplu este prezentat în figura 8.13, unde caracterele dispuse pe linii sînt citite pe coloane, ordinea coloanelor fiind dată de ordinea alfabetică a literelor unei chei.

cheie = POLIGRAF

text clar :

ordine = 76543812

```
AFOSTODA     AFOSTODATACANPOVESTIAFOSTCANICIO
TACANPOV
ESTIAFOS        text cifrat
TCANICIO
                DOOIAVSOTNAISAINOCTAFASCATETOPFC
```

Figura 8.13

8.4.6.Cifruri bloc

Criptografia a făcut un important salt odată cu introducerea calculatorului ca mijloc de cifrare şi descifrare. Dacă la cifrurile clasice secretul este asigurat, în principal prin folosirea unor chei de lungimi mari, la cele computaţionale accentul cade pe complexitatea algoritmilor de cifrare şi descifrare. Substituţiile şi transpoziţiile realizabile prin circuite foarte simple, sînt folosite în combinaţii care dau naştere unore **cifruri produs** greu de "spart", chiar de un criptanalist care dispune de mari cantităţi de text cifrat.

Un exemplu de obţinere a cifrurilor produs este prezentat în figura 8.14. El se bazează pe folosirea a două tipuri de circuite: circuit de permutare P (P-box), care realizează o transpoziţie a intrării şi circuit de substituţie S (S-box), care realizează o substituţie a intrării. In exemplul dat, circuitele P au 12 intrări şi 12 ieşiri, iar circuitele S au cîte trei. Intregul echipament cifrează blocuri de text clar de cîte 12 biţi, producînd blocuri de aceeaşi dimensiune de text cifrat.

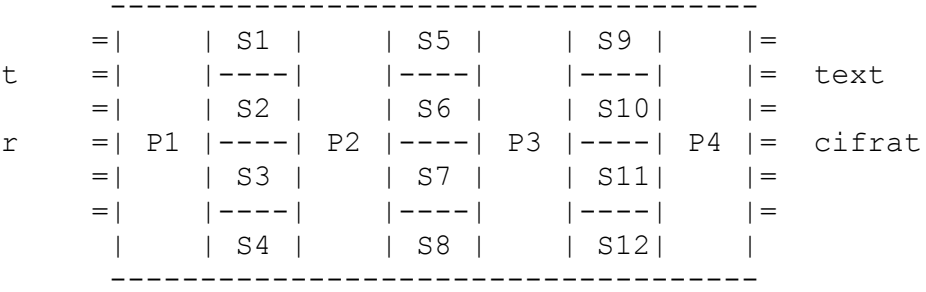


Figura 8.14

Exemplul cel mai cunoscut de cifrare bloc este sistemul DES (Data Encryption Standard), adoptat în 1977 în SUA (de National Bureau of Standards) pornind de la un cifru produs elaborat de IBM. In ciuda numeroaselor critici aduse standardului, el are o răspîndire mare, datorită şi implementărilor sale în forma unor circuite integrate pe scară largă, deosebit de rapide.

Schema generală a cifrării este prezentată în figura 8.15. Textul clar este cifrat în blocuri de cîte 64 de biţi, folosind o cheie de 56 de biţi (plus 8 de control la paritate). Transformările componente ale produsului cuprind următoarele:
- o transpoziţie iniţială a blocului de 64 de biţi;

- 16 iteraţii, parametrizate de chei diferite;
- un interschimb al celor două jumătăţi ale blocului (fiecare avînd 32 de biţi);
-o transpoziţie finală, care este inversa transpoziţiei iniţiale.

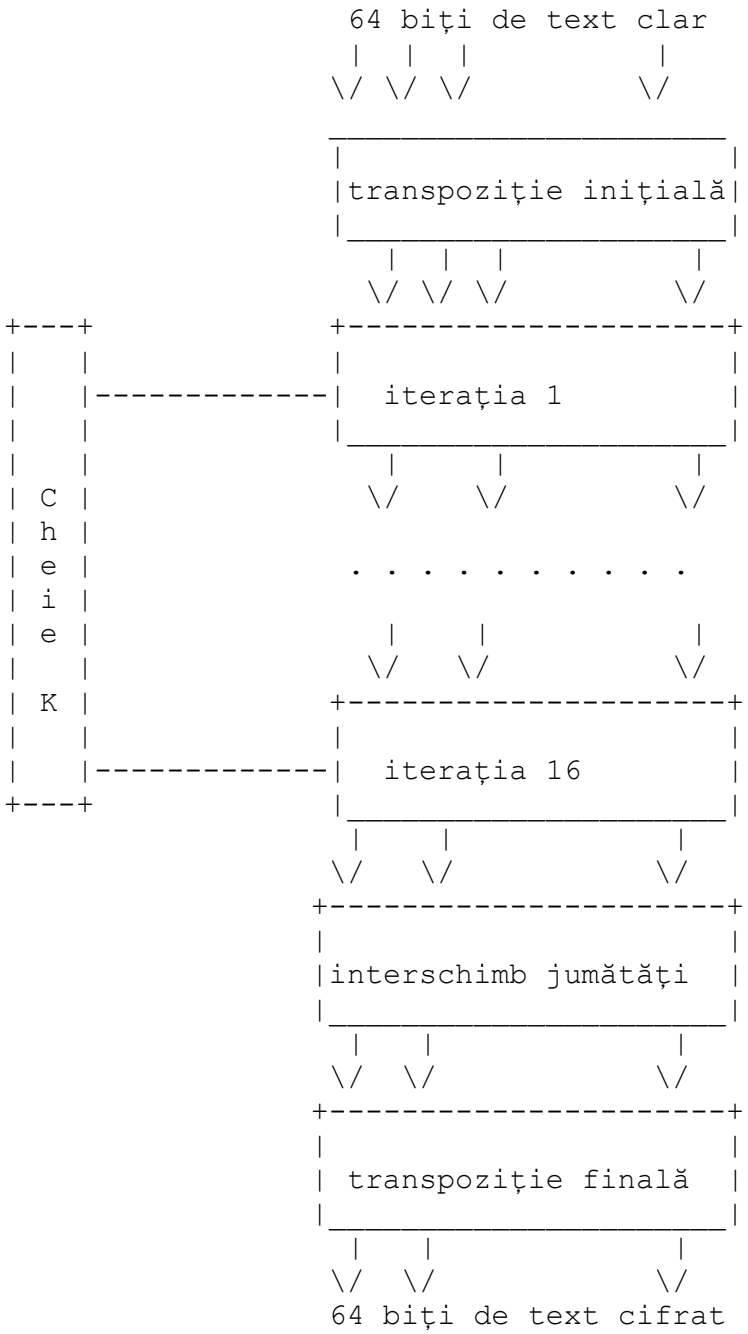


Figura 8.15.

Prelucrarea realizată la fiecare iteraţie "i" este ilustrată în figura 8.16. Ea constă din următoarele operaţii :
- intrarea este împărţită în jumătăţile stîngă L[i-1] şi dreaptă R[i-1];
- ieşirea L[i] este copia lui R[i-1], L[i] = R[i-1];

- ieşirea R[i] se obţine prin
$$R[i] = L[i-1] + f(R[i-1],K[i])$$
unde K[i] este o cheie de 48 de biţi, dependentă de cheia K a cifrului şi de numărul de ordine al iteraţiei, i.

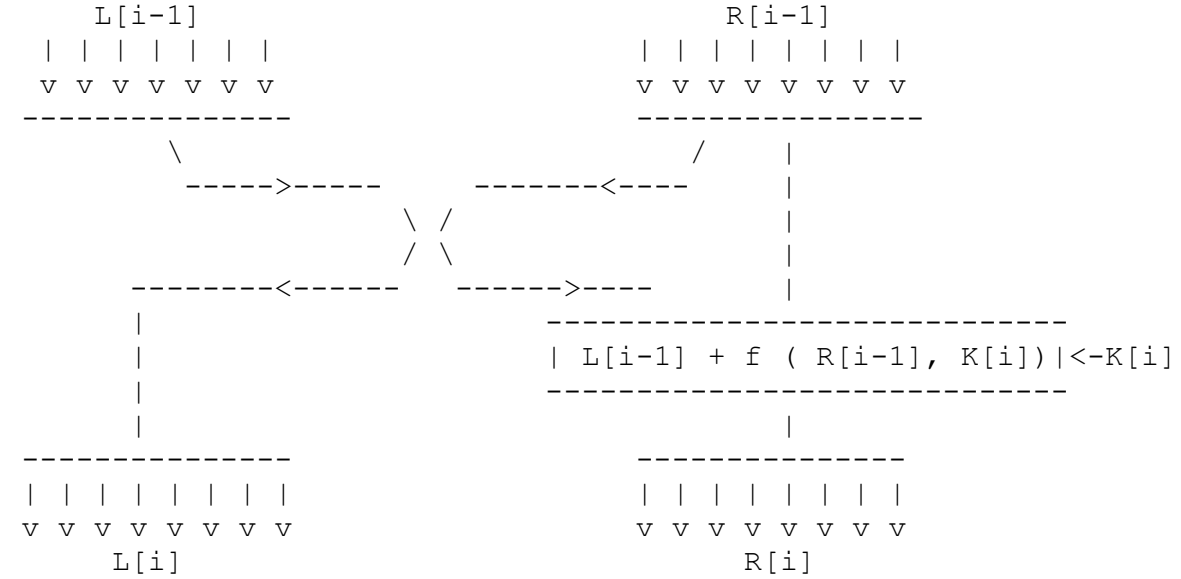


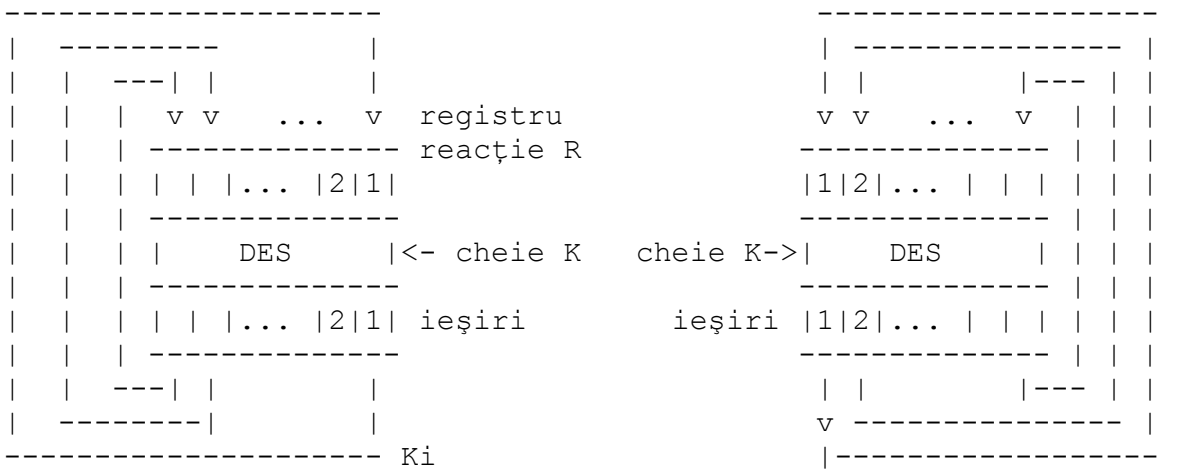
Figura 8.16.

Amănunte despre realizarea funcţiei f şi despre obţinerea cheilor folosite în cele 16 iteraţii sînt date în anexa 6, unde se prezintă un program de implementare a standardului.

8.4.7.Cifrarea secvenţială

Algoritmul DES poate fi adaptat pentru o funcţionare secvenţială. De altfel, standardul descrie patru moduri de operare, cifrarea pe bloc fiind doar unul dintre ele, (denumit ECB - Electronic Code Book). Ne referim în continuare la celelalte trei.

Funcţionarea ca **sistem secvenţial sincron cu reacţie bloc** (OFB - Output Fead Back) este ilustrată în figura 8.17.



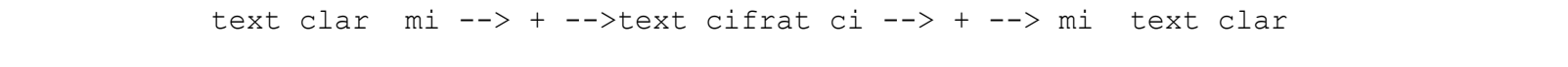


Figura 8.17

Textul cifrat se obţine combinînd prin operaţia sau-exclusiv unităţile de text clar m[i] cu cheia K[i]. Şirul cheilor K[i] este furnizat de un circuit DES, a cărui intrare este un registru de reacţie R. Pornind de la o valoare iniţială oarecare, conţinutul acestuia este actualizat la fiecare iteraţie, cu ieşirea circuitului DES. Lungimea cheii K[i] poate fi de unul sau mai multi biţi, cifrarea şi descifrarea putîndu-se face, de exemplu, la nivelul caracterelor mesajului.

Func?ionarea ca **sistem cu reacţie de text cifrat** (CFB - K-bit Cifer-Feed Back) este prezentată schematic în figura 8.18.

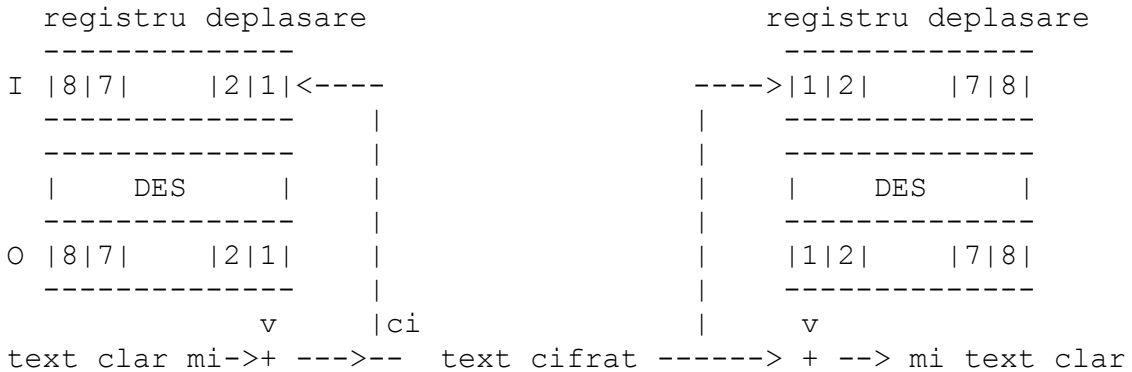


Figura 8.18

Fiecare circuit de cifrare/descifrare are un registru de deplasare, care păstrează blocul de intrare. La sursa, k biţi din textul clar sînt combinaţi prin operaţia sau-exclusiv cu k biţi ai ieşirilor circuitului (cei mai semnificativi), rezultatul fiind transmis destinatarului şi (totodată) introdus în registrul de deplasare I, în poziţiile mai puţin semnificative (restul biţilor se deplasează corespunzător). Ieşirea circuitului se schimbă, astfel că următorii k biţi de text clar se vor combina cu o altă valoare a cheii.

La recepţie fiecare k biţi primiţi sînt combinaţi, prin operaţia sau-exclusiv, cu k biţi ai ieşirii circuitului DES, refăcînd textul clar original. Rezultatul totodată introdus în poziţiile mai puţin semnificative ale registrului de deplasare. Evident, valorile iniţiale ale registrelor de intrare şi destinatarului trebuie să coincidă.

Exemplul dat ilustrează destul de bine caracteristicile cifrurilor secvenţiale, la care "cheia" folosită pentru cifrare, dependentă pe de o parte de cheia generală K a circuitului (care stabileşte dependenţa ieşirilor O de intrările I ale circuitului), este dependentă şi de istoria trecută a procesului de cifrare. Concret, valoarea ieşirii la un moment dat a circuitului DES depinde de

caracterele de text clar deja transmise, astfel că un acelaşi caracter din textul clar generează configuraţii diferite de biţi în textul cifrat. Şi acest mod de lucru se pretează la cifrarea succesivă a caracterelor unui mesaj.

Funcţionarea cu **înlănţuirea blocurilor cifrate** (CBC - Cipher Block Chaining) este schiţată în figura 8.19.

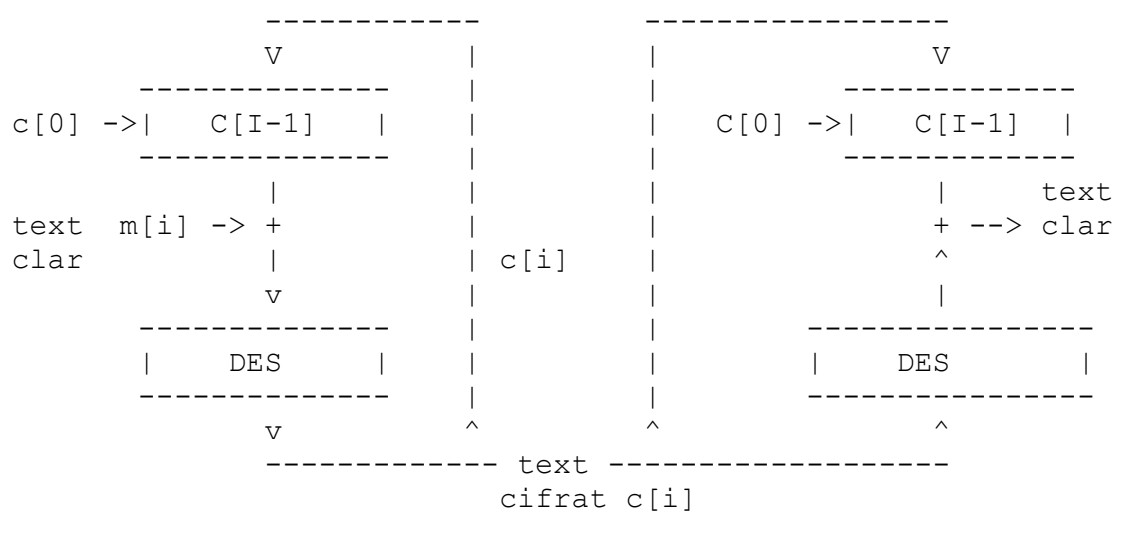


Figura 8.19

Textul cifrat c[i] este obţinut prin aplicarea la intrarea circuitului DES a unei valori care se obţine prin însumarea modulo 2 a unui bloc de text clar m[i] şi a ieşirii pasului anterior c[i-1] (valoarea iniţială c[0] este aplicată din exterior). Descifrarea lui c[i] se face aplicînd acest bloc la intrarea unui circuit DES şi însumînd modulo 2 rezultatul cu blocul c[i-1] de la pasul anterior.

8.4.8 Cifrarea prin funcţii greu inversabile

O prezentare a algoritmilor de cifrare nu poate omite cifrurile bazate pe funcţii greu inversabile. Menţionăm în continuare principalele idei pe care se bazează aceşti algoritmi, o tratare mai extinsă putînd fi găsită în [2].

Ideea de bază este relativ simplă şi, ca mai toate ideile mari, este inspirată din viaţă: este uşor de parcurs o stradă cu sens unic în sensul permis, dar, adeseori, imposibil în sens contrar. Similar, în cazul unei funcţii greu inversabile f, cunoscînd pe x este uşor de calculat f(x), în timp ce, calculul lui x din f(x) este foarte dificil. Utilizarea funcţiilor greu inversabile în criptografie are şi o mică adaptare: calculul lui x din f(x) trebuie să fie o problemă **intratabilă** doar pentru criptanalist, şi nu pentru destinatarul autorizat; în acest scop, acesta din urmă dispune de o informaţie suplimentară, de o **trapă** care face problema uşor de rezolavat. Prin convenţie, o problemă este

considerată intratabilă dacă nu există un algoritm de rezolvare care să opereze în timp polinomial.

Cele mai cunoscute cifruri din această categorie sînt cele bazate pe algoritmi exponenţiali şi pe problema rucsacului. In metodele PH (după numele autorilor, Pohling şi Hellman) şi RSA (Rivest, Shamir şi Adleman) cifrarea unui bloc de mesaj M (considerat ca o valoare numerică întreagă, cuprinsă între 0 şi n-1) se face prin calculul

$$C = (M \hat{} e) \bmod n$$

unde (e, n) reprezintă cheia de cifrare. Descifrarea se face utilizînd o cheie de descifrare (d,n) diferită de (e,n) astfel aleasă încît să fie respectată relaţia :

$$M = (C \hat{} d) \bmod n.$$

Condiţia necesară este ca

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

unde n este produsul a două numere prime foarte mari p şi q, $n = p*q$, F(n) este indicatorul lui Euler

$$F(n) = F(p) * F(q) = (p-1)(q-1),$$

iar $\max(p,q) < d < n$ este relativ prim cu n. Deoarece F(n) este greu calculabil,schema se poate utiliza în criptosisteme cu chei publice, în care se fac cunoscute e şi n, ţinîndu-se secret d.
In metoda MH (după autorii Merkle şi Hellman) se porneşte de la următoarea problemă (NP - completă) : fiind dat un număr natural C şi un vector A = (a1,a2,..an) de întregi pozitivi, să se găsească un subset al lui A a cărui sumă să fie egală cu C. Altfel spus, se cere determinarea lui X = (x1,x2,...xn) cu elemente binare, a.i.

$$C = \sum_{i=1,n} x[i] * a[i]$$

O soluţie x propusă poate fi uşor verificată prin cel mult n operaţii de adunare, în timp ce găsirea unei soluţii cere un număr de operaţii care creşte exponenţial cu n, făcînd problema netratabilă. Dacă A satisface proprietatea de dominanţă, adică

$$a[i] > \sum_{j=1,i-1} a[j]$$

atunci problema poate fi rezolvată simplu, în timp liniar. Ideea este folosită în criptare, în modul următor :
- un mesaj M, în reprezentare binară este cifrat prin calculul

$$C = A * M(T)$$

unde A este un vector "rucsac greu", reprezentînd cheia de cifrare publică; regăsirea lui M din C este o problemă netratabilă, pentru un criptanalist;
- receptorul autorizat dispune de o informaţie trapă secretă, cu ajutorul căreia transformă C şi A în C' şi A' astfel încît regăsirea lui M din C' şi A' reprezintă o problemă rucsac simplă.

8.4.9. Distribuirea cheilor

In sistemele simetrice cifrarea şi descifrarea se fac cu o aceeaşi cheie, care trebuie cunoscută atît de transmiţătorul cît şi de receptorul mesajelor. O problemă importantă a acestor sisteme este distribuirea cheilor în siguranţă. Pentru o organizaţie oarecare, generarea cheilor la un punct central şi distribuirea lor prin curier personal nu este realizabilă dacă numărul de utilizatori este mare. Există însă soluţii de a transmite cheile chiar prin reţea.

O variantă este utilizarea unui sistem ierarhic de chei. O organizaţie alege la întîmplare o cheie şi o transmite tuturor oficiilor, prin curier personal. Folosind această cheie (master), un grup de oficii situate într-o aceeaşi regiune pot stabili şi difuza între ele o cheie regională. Folosind această cheie, doi utilizatori din regiune pot stabili, de comun acord, cheia pe care o vor utiliza într-o sesiune de lucru. Deoarece cheia master şi cheile regionale sînt folosite rar, probabilitatea spargerii lor este foarte mică.

O altă variantă cere ca, la începutul unei conversaţii, un utilizator să furnizeze celuilalt un număr mare de criptograme, fiecare "ascunzînd" o cheie şi un număr de identificare a cheii. Corespondentul trebuie să aleagă o criptogramă, să o descifreze descoperînd astfel cheia (în acest sens poate primi informaţii care să scurteze operaţia) şi să transmită în clar primului utilizator numărul de identificare al cheii. Conversaţia poate decurge apoi în forma cifrată. Deoarece un intrus nu ştie care din numeroasele criptograme a fost aleasă, el trebuie să parcurgă un număr important din ele, pentru a o descoperi, ceea ce necesită un efort foarte mare.

8.4.10.Protecţia cheilor

In operaţii foarte importante, mai mulţi utilizatori pot împărţi o aceeaşi cheie, fiecare deţinînd o parte a ei. Cheia poate fi folosită doar dacă se pun împreună părţile tuturor membrilor grupului. Un algoritm ingenios (propus de Shamir în 1979) foloseşte un polinom P de gradul k-1, pentru un grup de k persoane. Coeficientul p[0] reprezintă cheia, ceilalţi alegînduse la întîmplare. Fiecărui membru al grupului i se dă o pereche de numere (x, P(x)), reprezentînd "partea sa de cheie".

Coeficienţii polinomului, deci şi cheia p[0], pot fi determinaţi dacă se cunosc cele k perechi de valori, prin rezolvarea unui sistem de ecuaţii liniare (necunoscutele fiind chiar aceşti coeficienţi). Calculele se fac modulo un număr prim, pentru a împiedica găsirea unor informaţii importante despre cheie folosind un număr mai mic de puncte.

KERBEROS

Kerberos este sistemul de autentificare folosit în proiectul Athena la MIT. El presupune că staţiile de lucru sînt nesigure şi cere clientului să se identifice ori de cîte ori cere un serviciu. Pentru a simplifica procedeul folosit la autentificare, Kerberos foloseşte următoarele idei:
- utilizatorul trebuie să se identifice o singură dată, la începutul unei sesiuni (prin nume şi parolă);
- parolele nu sînt niciodată transmise prin reţea în clar, ci doar criptate; mai mult, ele nu sînt păstrate în clar în staţiile de lucru;
- fiecare utilizator are o parolă şi fiecare serviciu are o parolă;
- singura entitate care cunoaşte toate parolele este serverul de autentificare.

Configuraţia pe care se bazează Kerberos este prezentată în figura 1. Funcţionarea cuprinde următorii paşi:

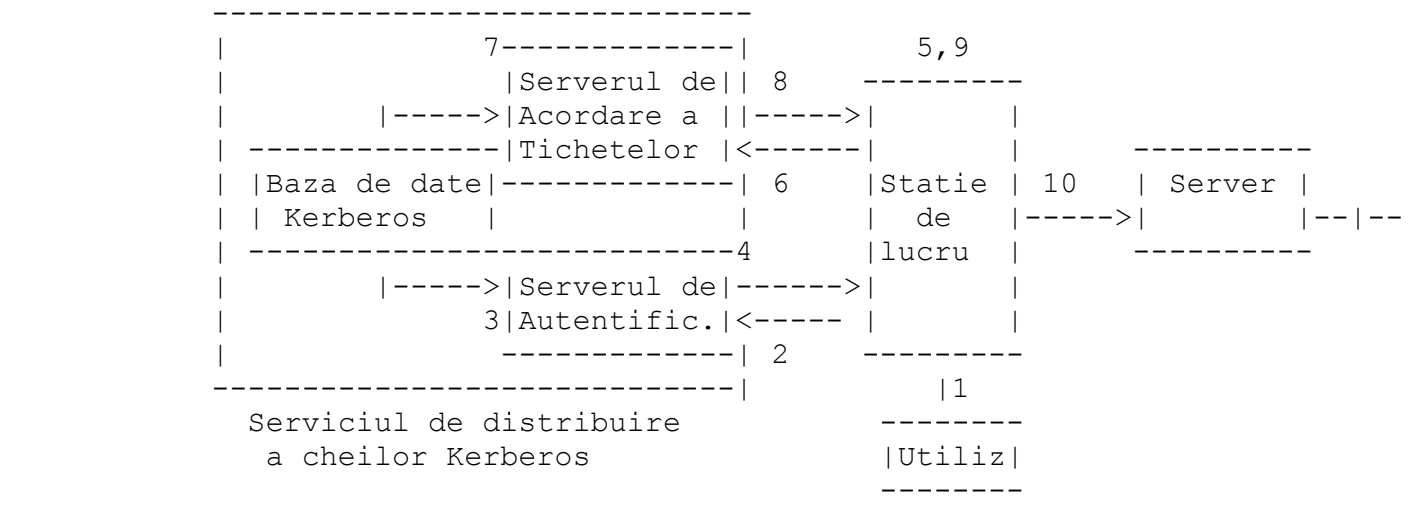


Figura 1

(1) Utilizatorul deschide o sesiune şi transmite numele (ca răspuns la **login**).

(2) Staţia transmite serverului de autentificare un mesaj care conţine numele de login şi un server particular de acordare a tichetelor (SAT):

mesaj = (nume_login, nume_SAT).

Mesajul nu este criptat, conţinînd nume ne-secrete.

(3) Serverul de autentificare (SA) caută în baza de date şi obţine:
- o cheie de criptare pentru utilizator;
- o cheie de criptare pentru SAT.

Ambele sînt parole criptate cu funcţii greu inversabile (one-way).

(4) SA trimite un răspuns staţiei. Răspunsul conţine:
- un tichet care garantează accesul la SAT;
- o cheie de sesiune SAT (un număr generat aleator de SA).

Tichetul are următoarea componenţă:
tichet= (nume_login, nume_SAT, adresa_retea_statie, cheie_sesiune_SAT).
El este criptat folosind cheia pentru SAT (de la pasul 3), obţinîndu-se un tichet **sigilat**.
Mesajul de răspuns, cu cele două componente este criptat folosind cheia utiizatorului. De notat că cheia de sesiune SAT apare de două ori în mesaj.

(5) Programul **login** primeşte mesajul şi cere parola utilizatorului. Parola este criptată printr-o funcţie greu inversabilă standard Unix, rezutatul fiind o cheie de cifrare a utilizatorului. Aceasta este folosită la decriptarea mesajului, parola clară fiind ştearsă din staţie. Staţia rămîne cu:
- un tichet sigilat;
- o cheie de sesiune SAT
pe care le memorează.

Folosirea unui serviciu reclamă obţinerea în prealabil a unui tichet corespunzător, de la SAT. Operaţiile următoare sînt executate automat şi transparent pentru utilizator.

(6) Staţia trimite SAT-ului un mesaj de forma
(tichet_sigilat, autentificator_sigilat, nume_serviciu).
Autentifiatorul este compus de staţie din:
(nume_login, adresa_reţea_staţie, timp_curent)şi este "sigilat" folosind cheia de sesiune SAT.

(7) SAT primeşte mesajul şi, mai întîi, decriptează tichetul sigilat, folosind cheia proprie de decriptare. De aici obţine cheia de sesiune SAT, cu care decriptează autentificatorul sigilat.
SAT verifică coincidenţa numelor din tichet şi autentificator şi numele SAT din tichet. El compară adresele de reţea din tichet, autentificator şi din mesajul primit. In fine, inspectează timpul curent din autentificator pentru a stabili dacă mesajul este recent.
SAT obţine din baza de date cheia serviciului indicat prin nume_serviciu.

(8) SAT formează o nouă cheie aleatoare de sesiune şi apoi creează un nou tichet:
tichet = (nume_login, nume_serviciu, adresa_reţea_staţie, cheie_sesiune_nouă)
care are aceeaşi componenţă cu cel de la pasul (4). Acest tichet este "sigilat" folosind cheia pentru serviciul cerut. SAT formează un mesaj de forma:
(cheie_sesiune_nouă, tichet_sigilat).
Mesajul este apoi "sigilat" folosind cheia de sesiune SAT (pe care staţia de lucu o cunoaşte) şi este transmis staţiei.

(9) Staţia primeşte mesajul sigilat şi îl decriptează folosind cheia de sesiune SAT. Din mesaj, preia tichetul sigilat, pe care

nu-l poate decripta, dar pe care-l poate trimite serviciului. El obţine totodată o nouă cheie de sesiune SAT pe care o foloseşte în următorul pas.

(10) Staţia construieşte un autentificator de forma
(nume_login, adresa_reţea_staţie, timp_curent)
şi îl sigilează folosind noua cheie de sesiune. Staţia poate transmite apoi un mesaj serviciului. Mesajul are acelaşi format cu cel transmis SAT-ului:
mesaj=(tichet_sigilat, autentificator_sigilat, nume_serviciu).
Mesajul nu este criptat, dar primele două componente sînt sigilate.

(11) Serviciul primeşte mesajul, decriptează tichetul sigilat folosind cheia sa de decriptare (pe care doar Kerberos o mai cunoaşte). Foloseşte apoi noua cheie de sesiune conţinută în tichet pentru a decripta autentificatorul, folosind acelaşi proces de validare care a fost descris la pasul (7).

Tichetele şi **autentificatorii** sînt punctele cheie pentru înţelegerea implementării schemei Kerberos.

(a) Pentru ca o staţie să folosească un serviciu este necesar un **tichet**. Tichetele se obţin de la SAT, cu excepţia primului pe care stţia îl ia de la SA.

(b) Tichetele deţinute de staţie nu sînt înţelese de ea, fiind criptate cu cheia serverului destinatar.
(c) Fiecare tichet este asociat cu o cheie de sesiune, asigurată la fiecare acordare a unui tichet.

(d) **Tichetele** sînt reutilizabile. Odată acordat unei staţii, tichetul poate fi refolosit o perioadă de timp (uzual 8 ore), după care expiră. Momentul acordării şi durata sînt, de asemenea, incluse în tichet.

(e) Un nou **autentificator** este cerut de fiecare dată cînd clientul iniţiază o nouă conexiune cu un server. Autentificatorul poartă momentul creării şi are o durată de valabilitate scurtă (minute).

(f) Un server păstrează istoria cererilor utilizatorului pentru care timpul din autentificator este încă valid. Aceasta permite depistarea dublurilor (şi deci a eventualelor fraude).

(g) Deoarece tichetul şi autentificatorul conţin adresa clientului, este destul de greu pentru o altă staţie să le folosească (ar trebui ca ea să-şi schimbe adresa!)

(h) Odată validată cererea de serviciu, clientul şi serverul folosesc o cheie privată (cunoscută în plus doar de Kerberos). Utilizarea cheii este opţională.

(i) Pe parcursul unei sesiuni, o listă de

(nume_server, tichet_sigilat, cheie_sesiune)
este păstrată de staţia de lucru în numele utilizatorului. Lista nu poate fi folosită de altcineva, numele utilizatorului fiind criptat.Lista este distrusă la terminarea sesiunii.

(j) Odată ce un server a validat un client, clientul poate cere validarea serverului, evitînd astfel ca un intrus să joace rolul serverului. Clientul cere ca serverul să transmită un mesaj constînd din timpul curent conţinut de autentificatorul clientului, modificat cu o unitate. Mesajul este criptat folosind cheia de sesiune transmisă de client serverului. Un intrus nu cunoaşte cheia de criptare a serverului, deci nu poate obţine cheia de sesiune din tichetul sigilat sau timpul din autentificatorul sigilat.